EECE 5639 Computer Vision I

Lecture 21

- **Dynamics-based Tracking; SFM**
- **Project 4 is out. Due April 12.**
- **Hw 5 is out. Due April 15**

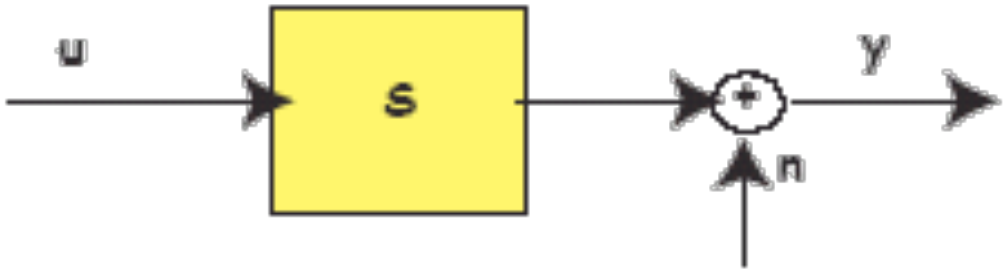
Next Class

- **SFM**
- **Object Recognition**

Tracking Using Tokens

Dynamical System

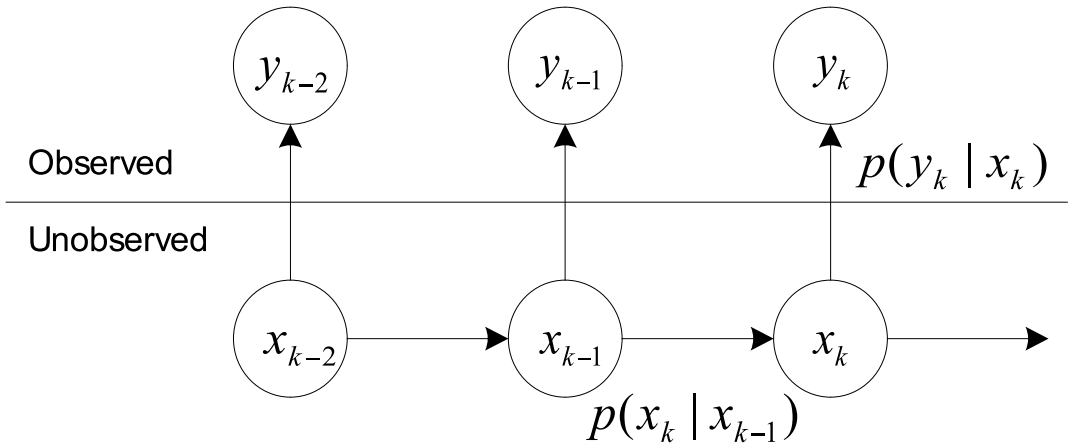
*A **dynamical system** is a mathematical formalization for any fixed "rule" which describes the time dependence of a point's position in its ambient space.*



- A dynamical system has a state determined by a collection of numbers.
- The evolution rule describes what future states follow the current state.

Dynamic State Space Model

The state x_k, the observation y_k at time k



- Assumptions
  - The current state depends only on n past states.

$$p(x_k|x_{k-1}, \ldots, x_{-\infty}) = p(x_k|x_{k-1}, \ldots, x_{k-n})$$

  - The observations are independent, given states

$$p(y_i, y_j, \ldots, y_k|x_i) = p(y_i|x_i)p(y_j, \ldots, y_k|x_i)$$

Recursive Filters

Main Idea:

Update the estimate based on the LAST measurement, instead of recomputing it from the beginning.

Example: Recursive Average

Estimate a CONSTANT x based on k measurements:

$y_i = x + v_i \quad i = 1, \ldots, k$

Non-recursive:

$$\hat{x}_k = \frac{1}{k} \sum_{i=1}^{k} y_i$$

Recursive:

$$\hat{x}_k = \frac{1}{k}[(k-1)\hat{x}_{k-1} + y_k]$$

Example: Recursive Average

Recursive estimate of a CONSTANT x based on k measurements:

$$\hat{x}_k = \frac{1}{k}[(k-1)\hat{x}_{k-1} + y_k]$$

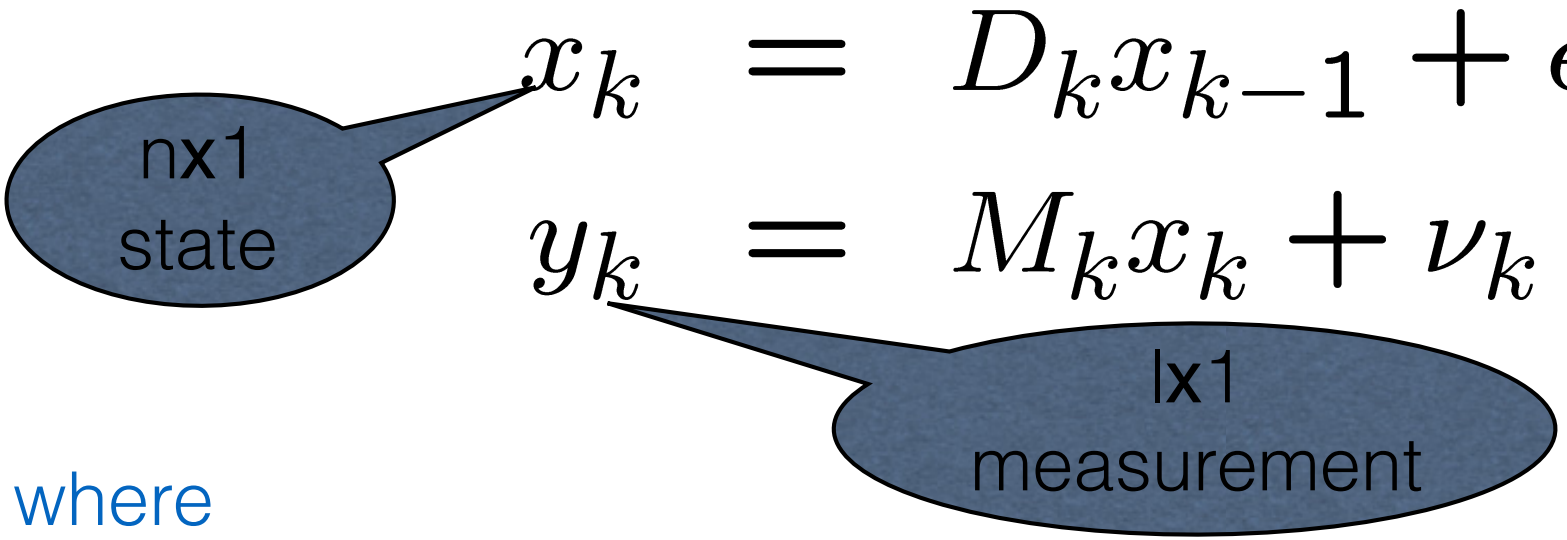
$$\hat{x}_k = \hat{x}_{k-1} + \frac{1}{k}[y_k - \hat{x}_{k-1}]$$

Previous estimate

Measurement Residual

Discrete KALMAN Filter

Consider a LINEAR DYNAMIC system:



$$x_k = D_k x_{k-1} + \epsilon_{k-1}$$

$$y_k = M_k x_k + \nu_k$$

where

$$\epsilon_k \sim N(0, \Sigma_{dk}) \quad \nu_k \sim N(0, \Sigma_{mk})$$

are UNCORRELATED noise

Discrete Kalman Filter

Notation:

Estimate BEFORE measurement k: $\hat{x}_k(-)$

Estimate AFTER measurement k: $\hat{x}_k(+)$

Estimate ERROR BEFORE measurement k:

$$\tilde{x}_k(-) = \hat{x}_k(-) - x_k$$

Estimate ERROR AFTER measurement k:

$$\tilde{x}_k(+) = \hat{x}_k(+) - x_k$$

Discrete Kalman Filter

Want a LINEAR recursive update equation:

$$\hat{x}_k(+) = K'_k \hat{x}_k(-) + K_k y_k$$

Discrete Kalman Filter

We want to find K'_k and K_k so that the estimate below is unbiased and min. variance.

$$\hat{x}_k(+) = K'_k \hat{x}_k(-) + K_k y_k$$

Unbiased ,

$$E[\tilde{x}_k(+)] = E[\tilde{x}_k(-)] = 0$$

Using

$$\tilde{x}_k(-) = \hat{x}_k(-) - x_k$$

$$\tilde{x}_k(+) = \hat{x}_k(+) - x_k$$

$$\hat{x}_k(+) = K'_k \hat{x}_k(-) + K_k y_k$$

$$y_k = M_k x_k + \nu_k$$

$$\tilde{x}_k(+) = K'_k[\tilde{x}_k(-) + x_k] +$$

$$K_k[M_k x_k + \nu_k] - x_k$$

$$\tilde{x}_k(+) = K'_k[\tilde{x}_k(-) + x_k] +$$

$$K_k[M_k x_k + \nu_k] - x_k$$

$$= (K'_k + K_k M_k - I)x_k +$$

$$K'_k \tilde{x}_k(-) + K_k \nu_k$$

$$\tilde{x}_k(+) = (K'_k + K_k M_k - I)x_k + K'_k \tilde{x}_k(-) + K_k \nu_k$$

Imposing that for every $\mathbf{x}_k$ $\quad E[\tilde{x}_k] = 0$

$$(K'_k + K_k M_k - I) = 0$$

$$\boxed{K'_k = I - K_k M_k}$$

Then the estimate is given by:

$$\hat{x}_k(+) = [I - K_k M_k]\hat{x}_k(-) + K_k y_k$$

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k[y_k - M_k\hat{x}_k(-)]$$

Previous estimate

Measurement Residual

And the estimation error is given by:

$$\tilde{x}_k(+) = [I - K_k M_k]\tilde{x}_k(-) + K_k \nu_k$$

Min Error Covariance:

$$\min \text{trace} P_k(+) = \text{trace} E[\tilde{x}_k(+)\tilde{x}_k(+)^T]$$

$$\tilde{x}_k(+) = [I - K_k M_k]\tilde{x}_k(-) + K_k \nu_k$$

$$\tilde{x}_k(+)\tilde{x}_k(+)^T = \{[I-K_k M_k]\tilde{x}_k(-)+K_k\nu_k\}\{\tilde{x}_k(-)^T[I-K_k M_k]^T+\nu_k^T K_k^T\}$$

$$E[\tilde{x}_k\tilde{x}_k^T] = P_k \qquad E[\nu_k {\nu_k}^T] = \Sigma_{mk} \qquad E[\nu_k\tilde{x}_k^T] = 0$$

$$P_k(+) = [I-K_k M_k]P_k(-)[I-K_k M_k]^T+K_k\Sigma_{mk}K_k^T$$

Using that for B symmetric $\dfrac{\partial \text{trace}(ABA^T)}{\partial A} = 2AB$

$$P_k(+) = [I - K_k M_k] P_k(-) [I - K_k M_k]^T + K_k \Sigma_{m_k} K_k^T$$

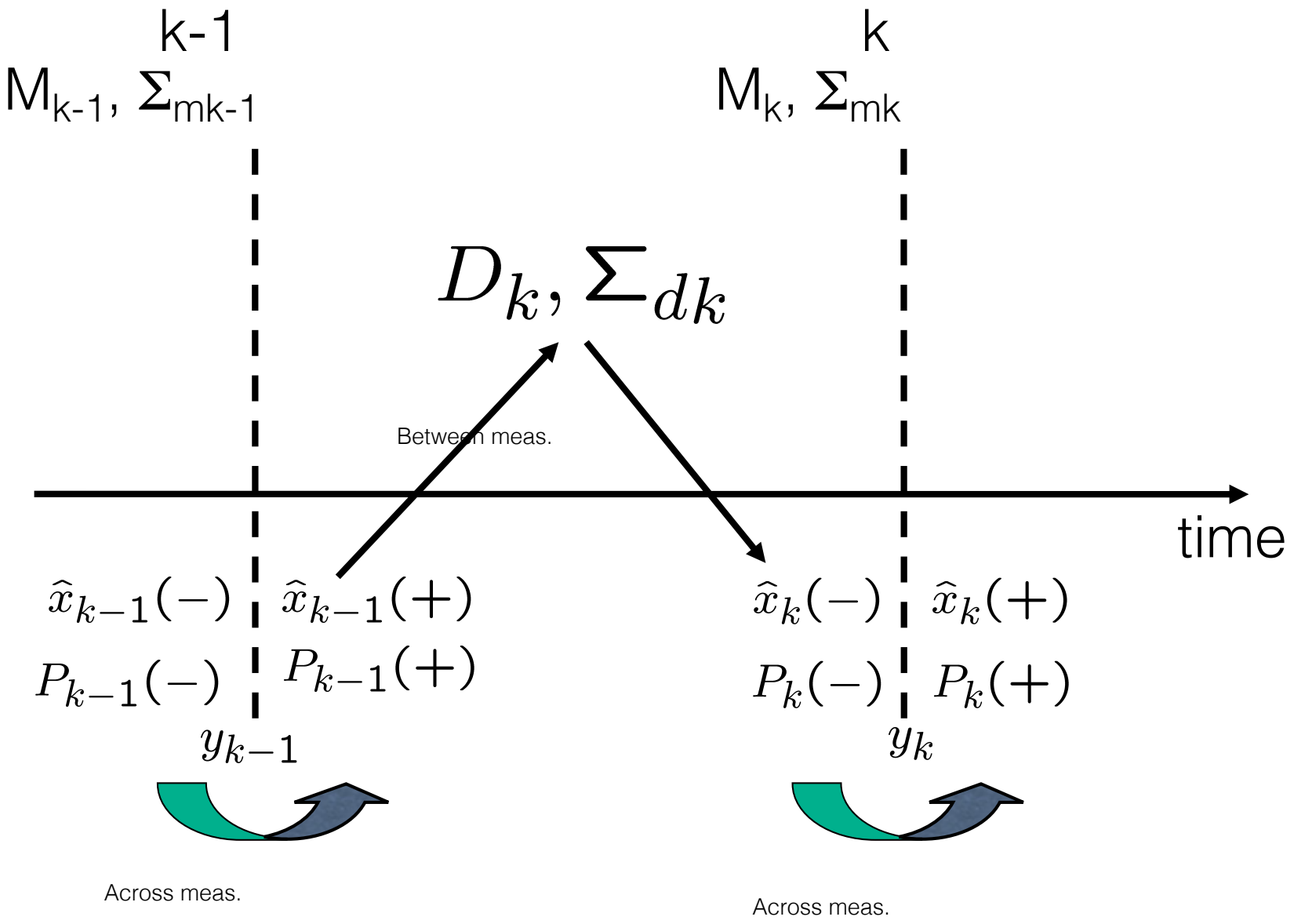
$$\frac{\partial \text{trace} P_k(+)}{\partial K_k} = 2[I - K_k M_k] P_k(-) + 2 K_k \Sigma_{m_k} = 0$$

$$K_k = P_k(-) M_k^T [M_k P_k(-) M_k^T + \Sigma_{m_k}]^{-1}$$

$$P_k(+) = [I - K_k M_k] P_k(-)$$

OK ... that was a lot of algebra! How do we use it?

Timing Diagram



Kalman Filter Algorithm

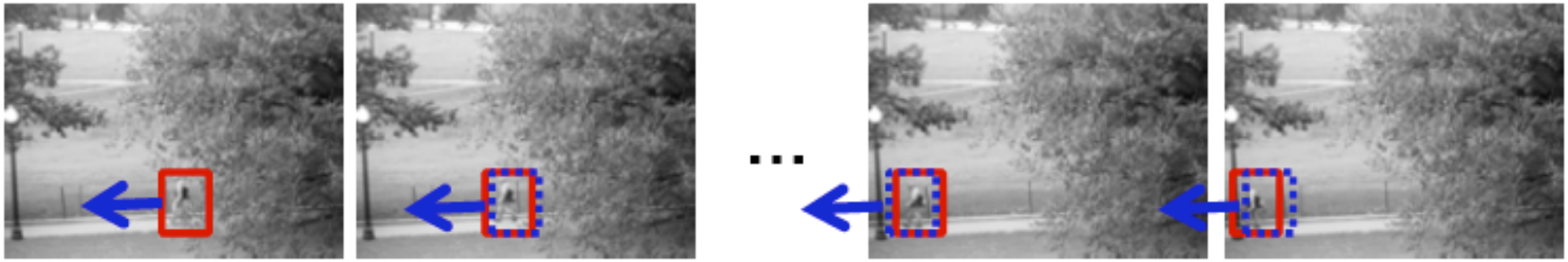
Between Measurements:

$$\begin{aligned} \hat{x}_k(-) &= D_k \hat{x}_{k-1}(+) \\ P_k(-) &= \Sigma_{dk} + D_k P_{k-1}(+) D_k \end{aligned}$$

Across Measurements:

$$\begin{aligned} K_k &= P_k(-) M_k^T [M_k P_k(-) M_k^T + \Sigma_{mk}]^{-1} \\ \hat{x}_k(+) &= \hat{x}_k(-) + K_k [y_k - M_k \hat{x}_k(-)] \\ P_k(+) &= [I - K_k M_k] P_k(-) \end{aligned}$$

Computer Vision Example



Time t Time t+1

Use dynamics to PREDICT and ESTIMATE the position of the target.

Advantages:

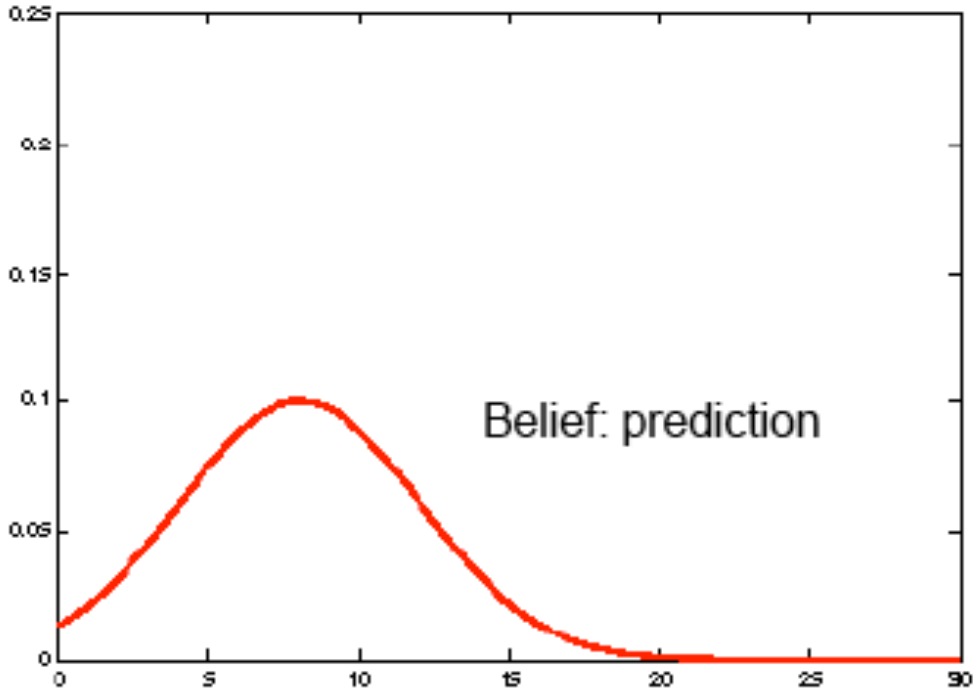
Reduces search space for the target

Improves the estimates of the location since measurements are noisy

Tracking using Prediction

Time t

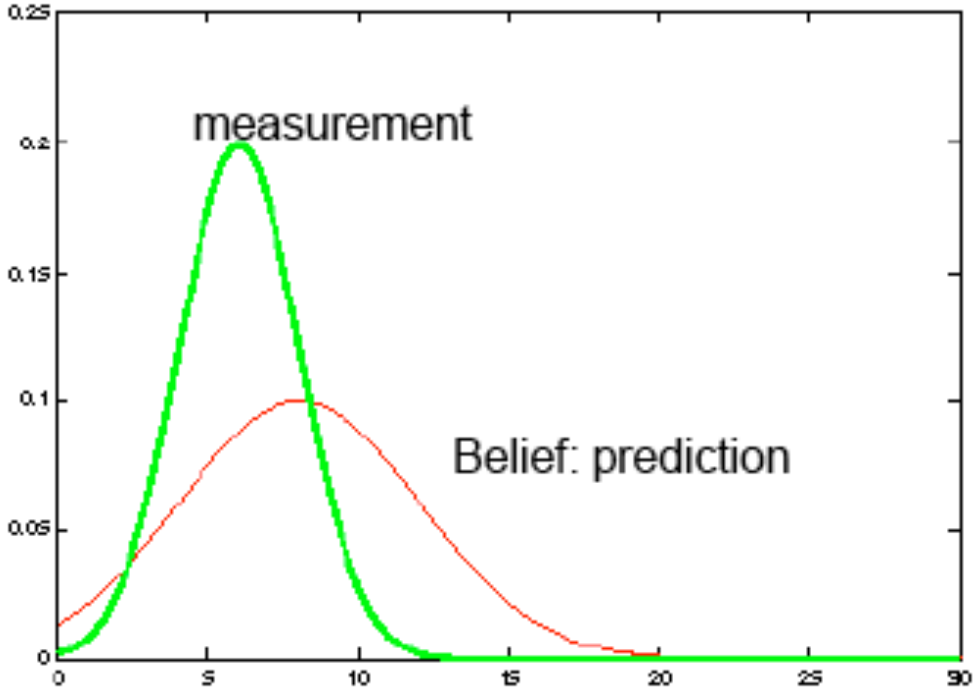
Time t+1



Tracking using Prediction

Time t

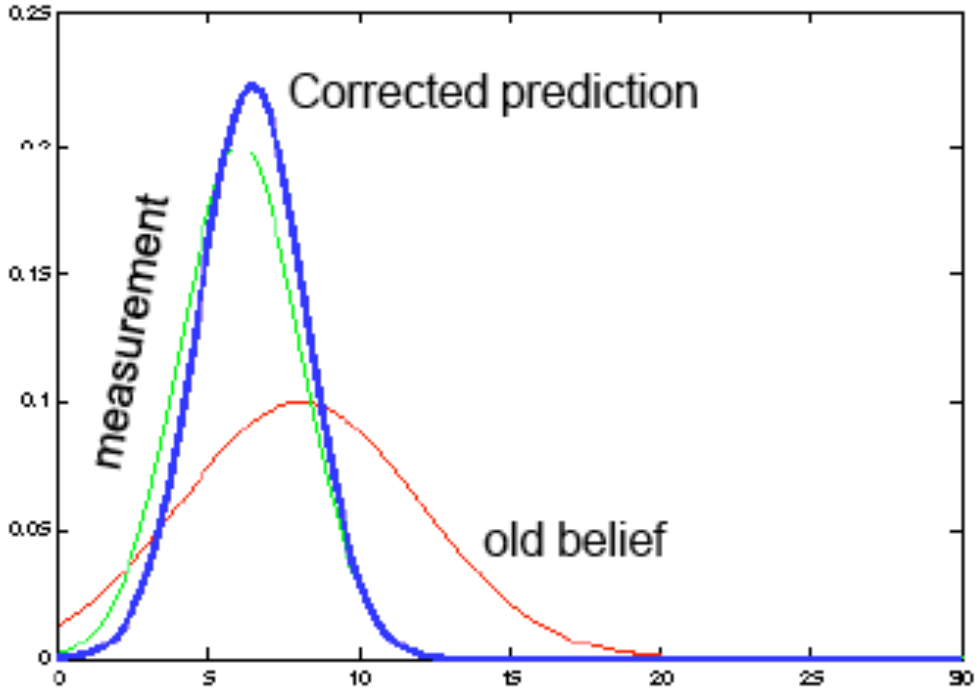
Time t+1



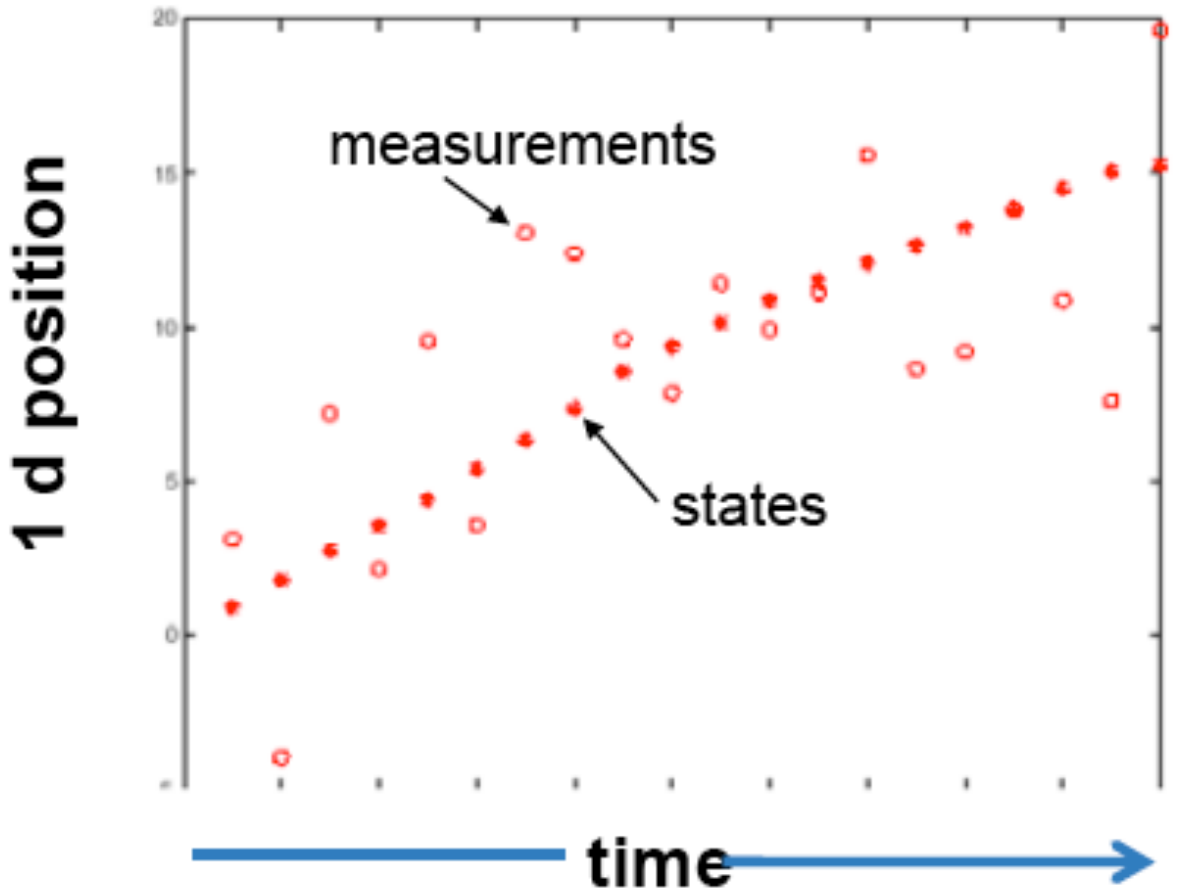
Tracking using Prediction

Time t

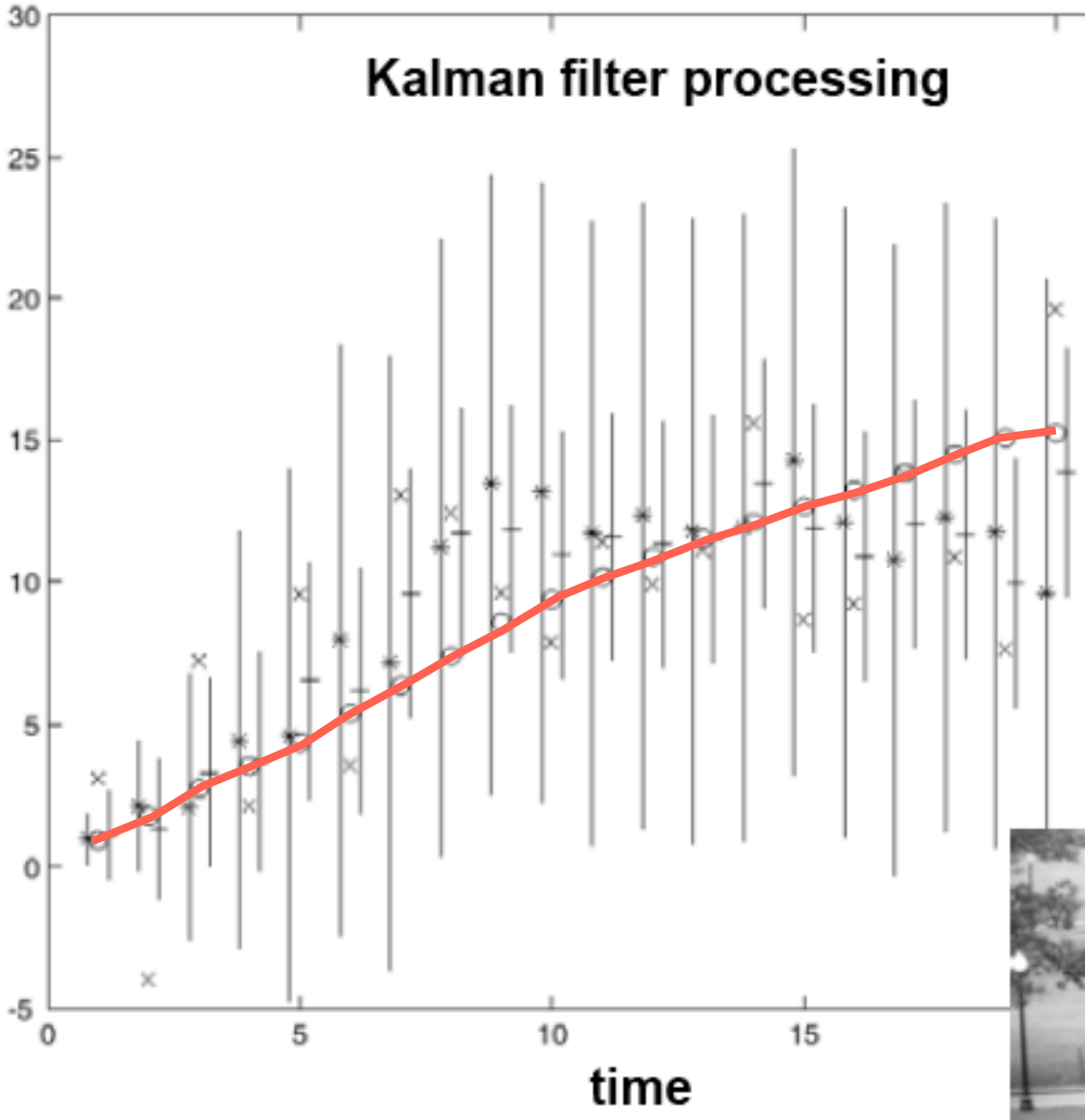
Time t+1



Example: Constant Velocity (1D points)



Constant Velocity Model



o state

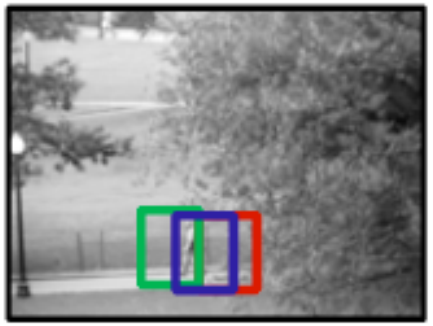
x measurement

* predicted mean estimate

\+ corrected mean estimate

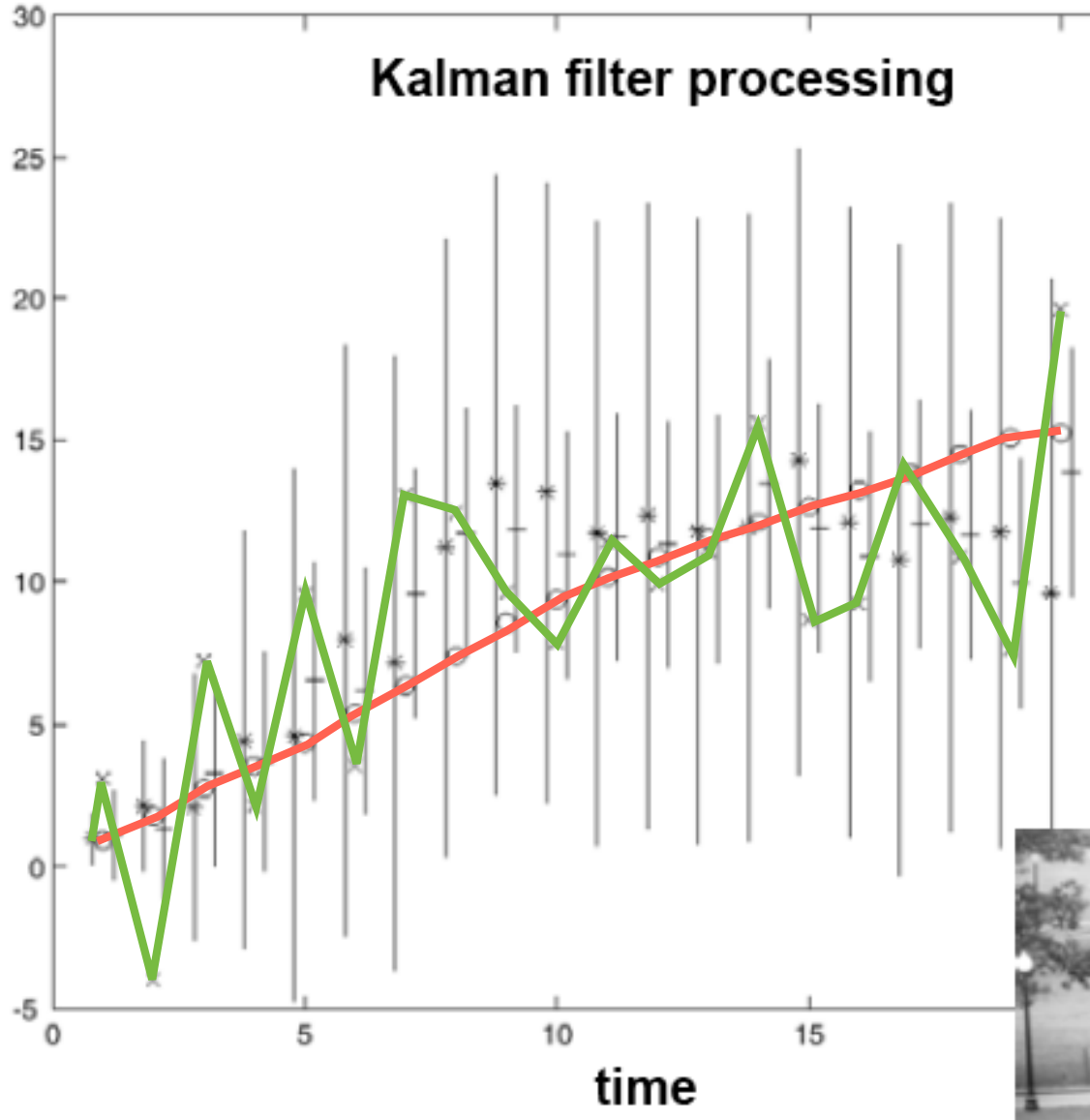
bars: variance estimates before and after measurements

Time t



Time t+1

Constant Velocity Model



o state

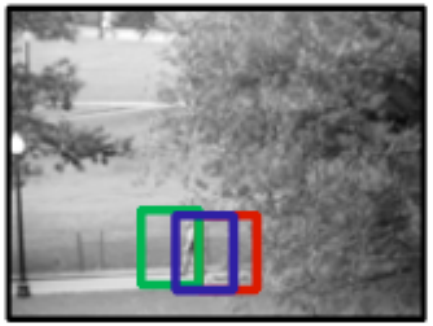
x measurement

* predicted mean estimate

\+ corrected mean estimate

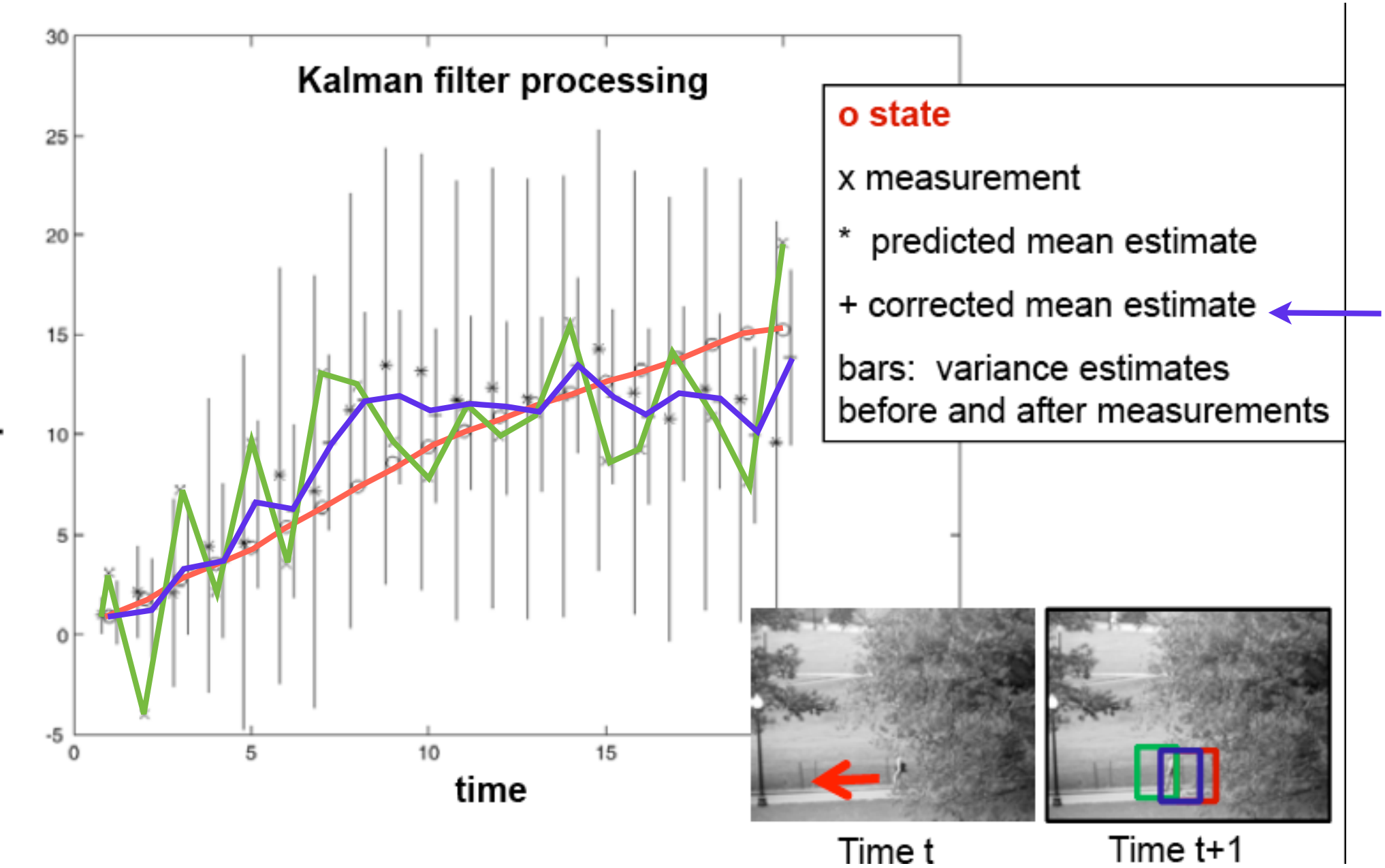
bars: variance estimates before and after measurements

Time t



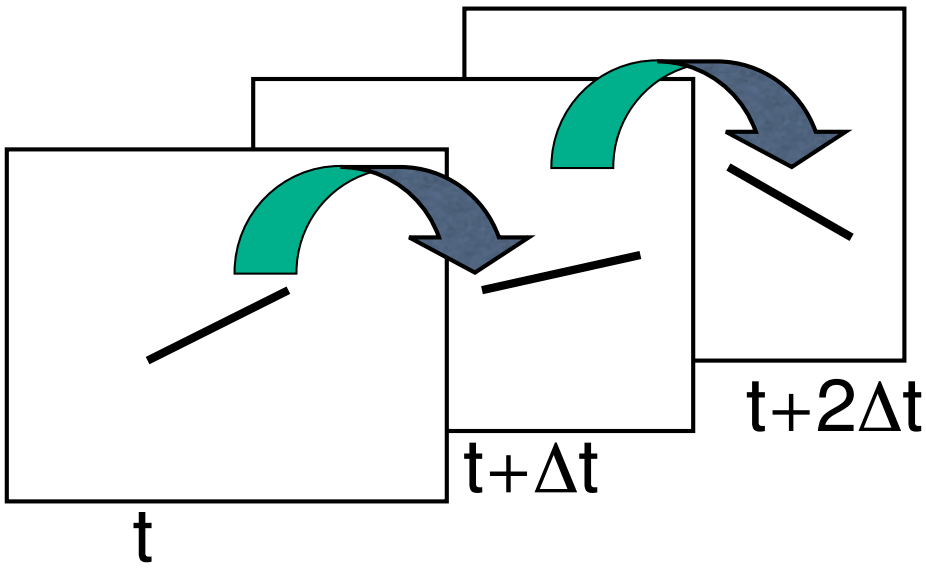
Time t+1

Constant Velocity Model

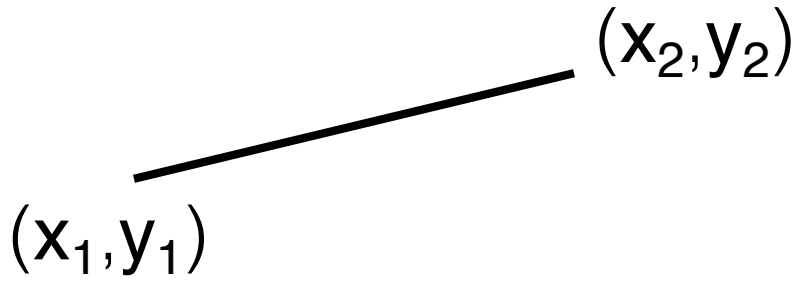


Computer Vision Example

2D Token Tracking: tracking an image segment across frames



A segment can be represented by 4 parameters:



Tracking a Segment

Represent a segment with a vector:

$$r = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix}_{4 \times 1}$$

Define the "state" vector:

$$x = \begin{bmatrix} r \\ \dot{r} \\ \ddot{r} \end{bmatrix}_{12 \times 1}$$

At each time **t**, we measure **r**

$$r_k = \begin{bmatrix} I_4 & 0 & 0 \end{bmatrix} x_k + \nu_k$$

Where do I get the dynamics?

Prior information, modeling, identification

Common assumptions:

- “random” walk
- constant velocity
- constant acceleration

Tracking a Segment

We need the dynamics. Assume for ex. CONSTANT acceleration:

$$x_k = \begin{bmatrix} I_4 & I_4\Delta t & \frac{1}{2}I_4\Delta t^2 \\ 0 & I_4 & I_4\Delta t \\ 0 & 0 & I_4 \end{bmatrix} x_{k-1} + \epsilon_k$$

$$\hat{x}_o = \begin{bmatrix} r_o \\ 0 \\ 0 \end{bmatrix} \quad P_o = \begin{bmatrix} \Sigma_{mo} & 0 & 0 \\ 0 & \lambda^2 I_4 & 0 \\ 0 & 0 & \lambda^2 I_4 \end{bmatrix}$$

Tracking Without Assuming Dynamics

"simple" dynamics might fail if there is prolonged occlusion

We can use tools from system identification (Hankel matrix) to predict future measurements without assuming a dynamic model.

Capturing Dynamics from Experimental Data: The Hankel Matrix

Given a sequence of measurements of d-dimensional vectors:

$$y_1, y_2, y_3 \cdots$$

Its Hankel matrix is defined as:

$$H_y = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n-1} & y_n & \cdots & y_p \\ y_2 & y_3 & \cdots & y_n & y_{n+1} & \vdots & y_{p+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_m & y_{m+1} & \cdots & y_{m+n-2} & y_{m+n-1} & \cdots & y_{m+p-1} \end{bmatrix}$$

md x p

Rank(H_y) measures the complexity of the underlying dynamics: after we have "enough" measurements the rank of the Hankel matrix does not increase.

Using a Regressor

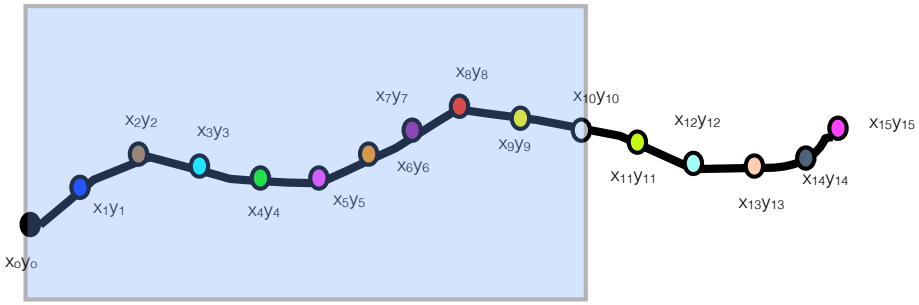
Model the dynamics using a simple regressor:

Regressor:

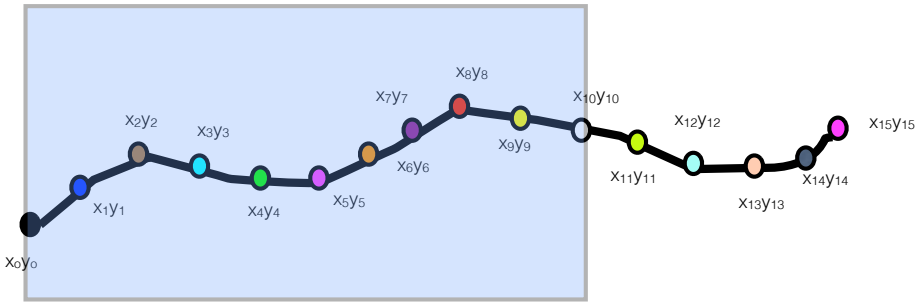
$$y_k = \sum_{i=1}^{\text{Rank}(H_y)} a_i y_{k-i}$$

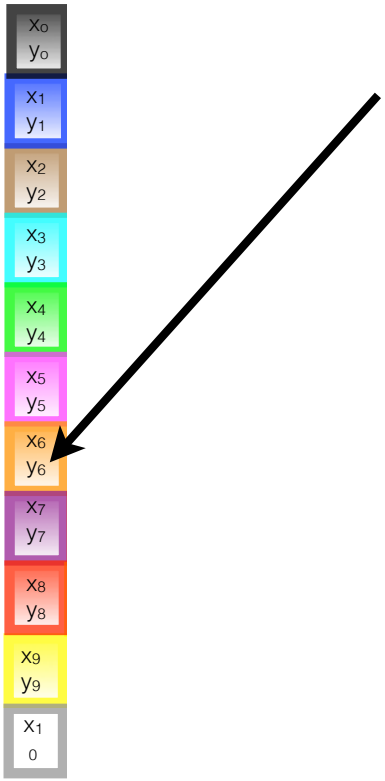
$$H_y = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n-1} & y_n \\ y_2 & y_3 & \cdots & y_n & y_{n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_m & y_{m+1} & \cdots & y_{m+n-2} & y_{m+n-1} \end{bmatrix}$$

Hankel Matrix



Hankel Matrix



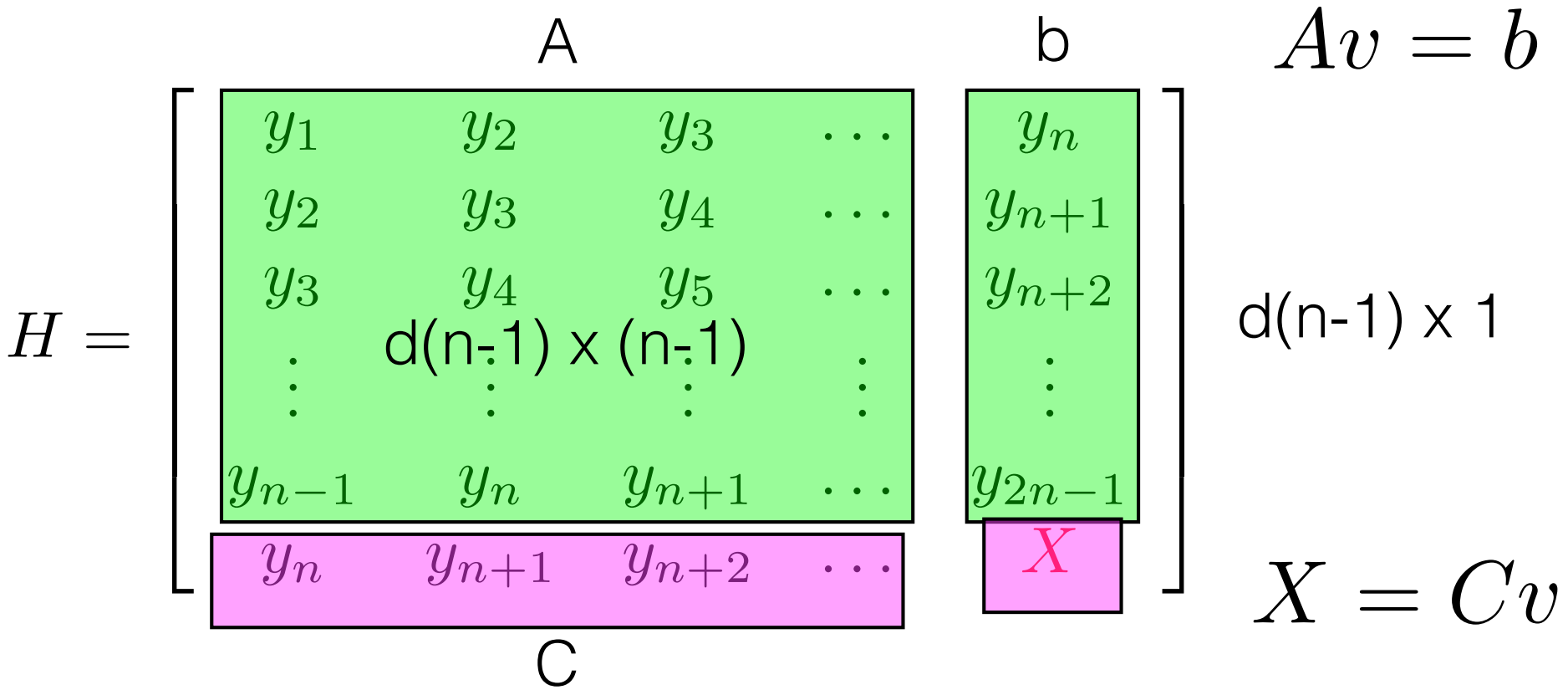


$$\sum_{i=1}^{\text{rank}(H)} a_i \begin{bmatrix} x_{k-i} \\ y_{k-i} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

$$\text{Rank}(H) = n = \text{Complexity of Dynamics}$$

Predicting the next measurement

The new measurement should not increase the complexity of the dynamics:



The last column must be a linear combination of the previous ones. **(But we should know the order of the system n-1).**

Data Prediction: Receding Horizon Tracking

Particle Kalman RHRM

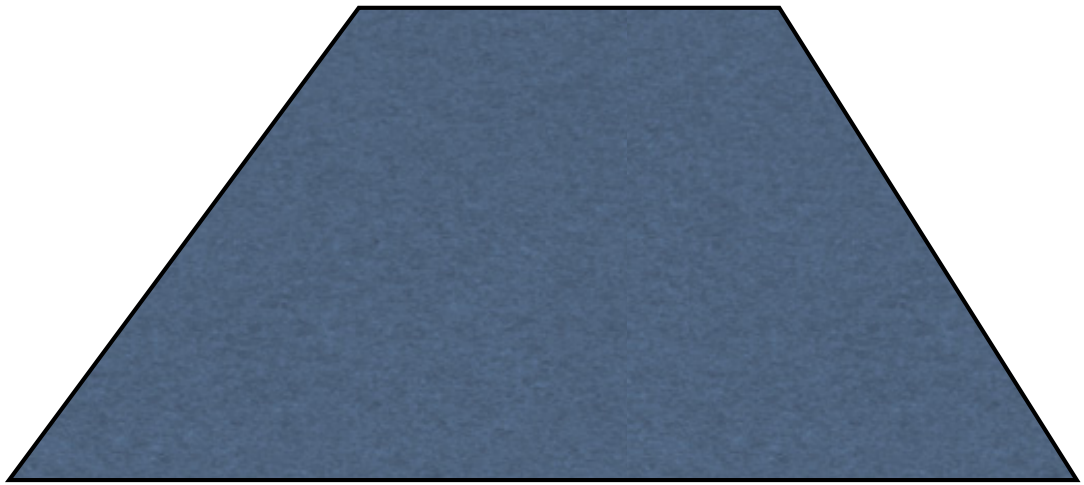
Hankel approach:

- Assemble Hankel matrix with noise

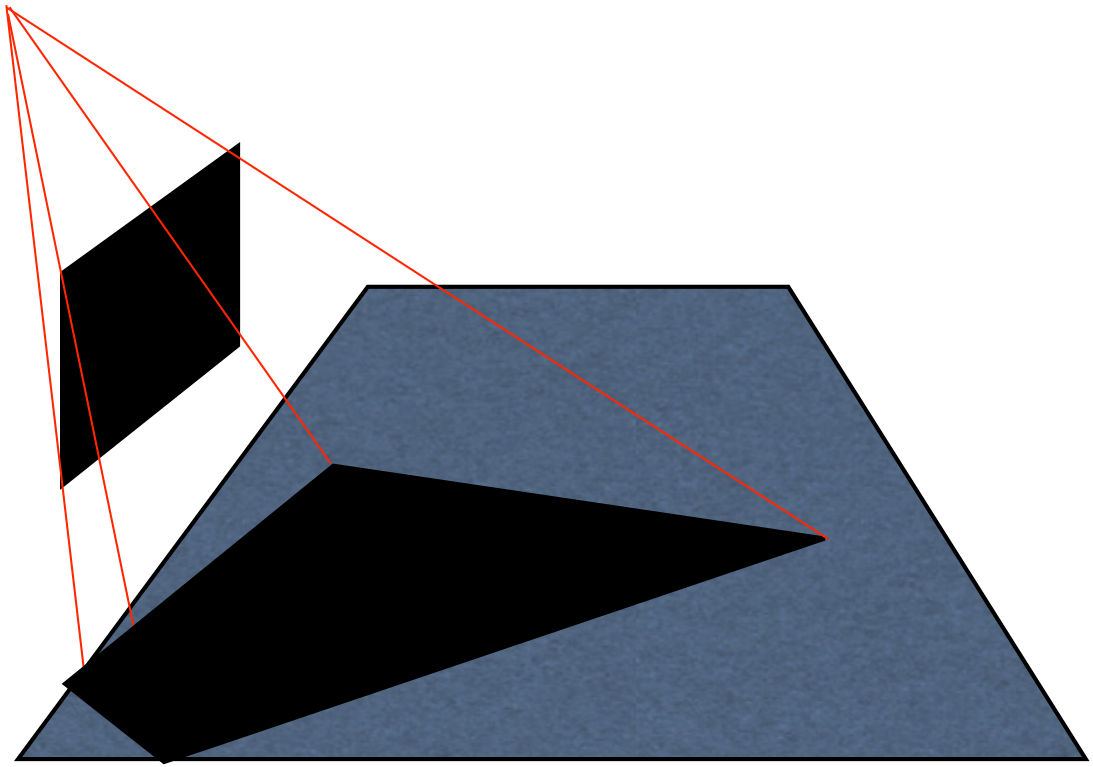
$$H = \begin{bmatrix} y_1+\eta_1 & y_2+\eta_2 & y_3+\eta_3 & \cdots & y_n+\eta_n \\ y_2+\eta_2 & y_3+\eta_3 & y_4+\eta_4 & \cdots & y_{n+1}+\eta_{n+1} \\ y_3+\eta_3 & y_4+\eta_4 & y_5+\eta_5 & \cdots & y_{n+2}+\eta_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_n+\eta_n & y_{n+1}+\eta_{n+1} & y_{n+2}+\eta_{n+2} & \cdots & y_{2n-1}+\eta_{2n-1} \end{bmatrix}$$

- Minimize rank(H) wrt noise
- Predict next measurement/update

Surveillance of Large Public Spaces

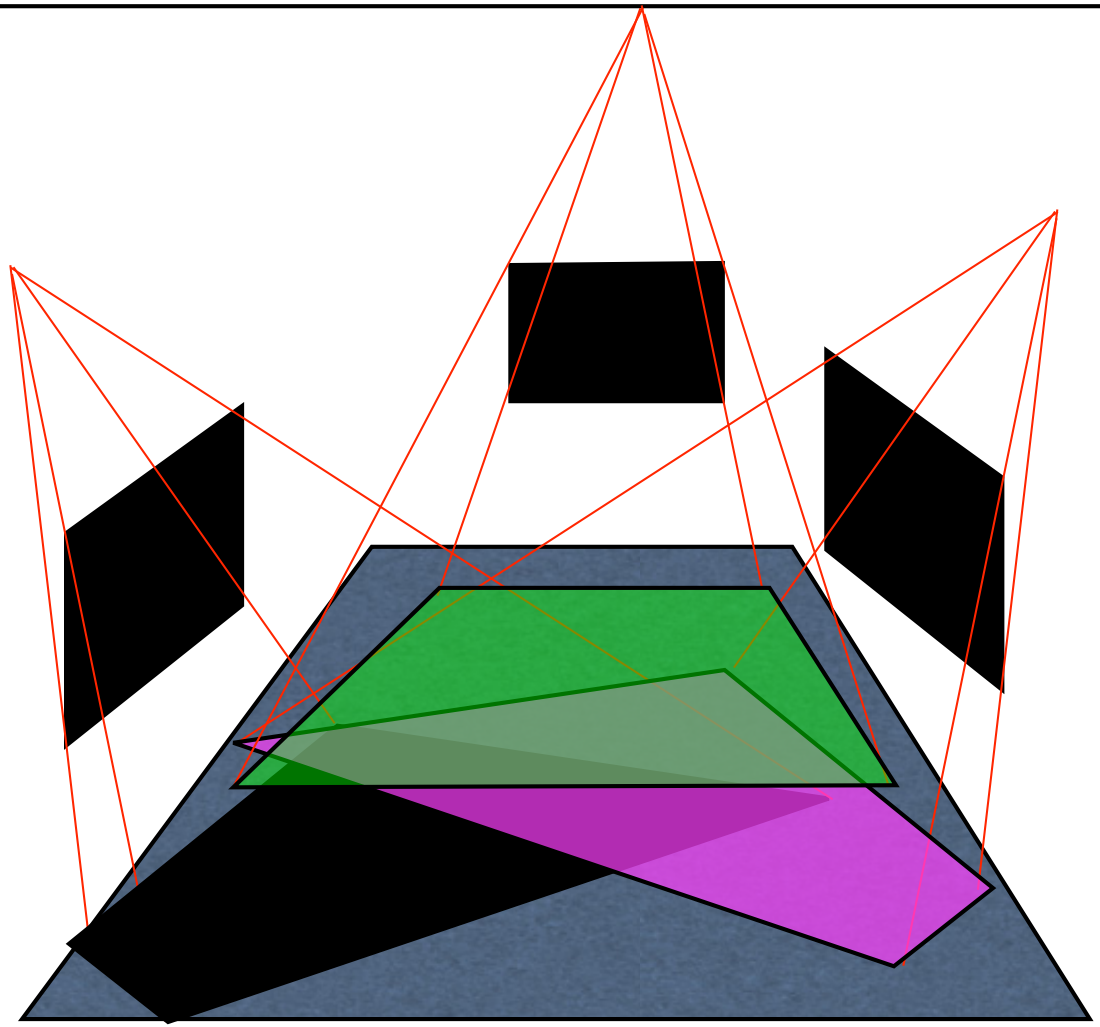


Surveillance of Large Public Spaces



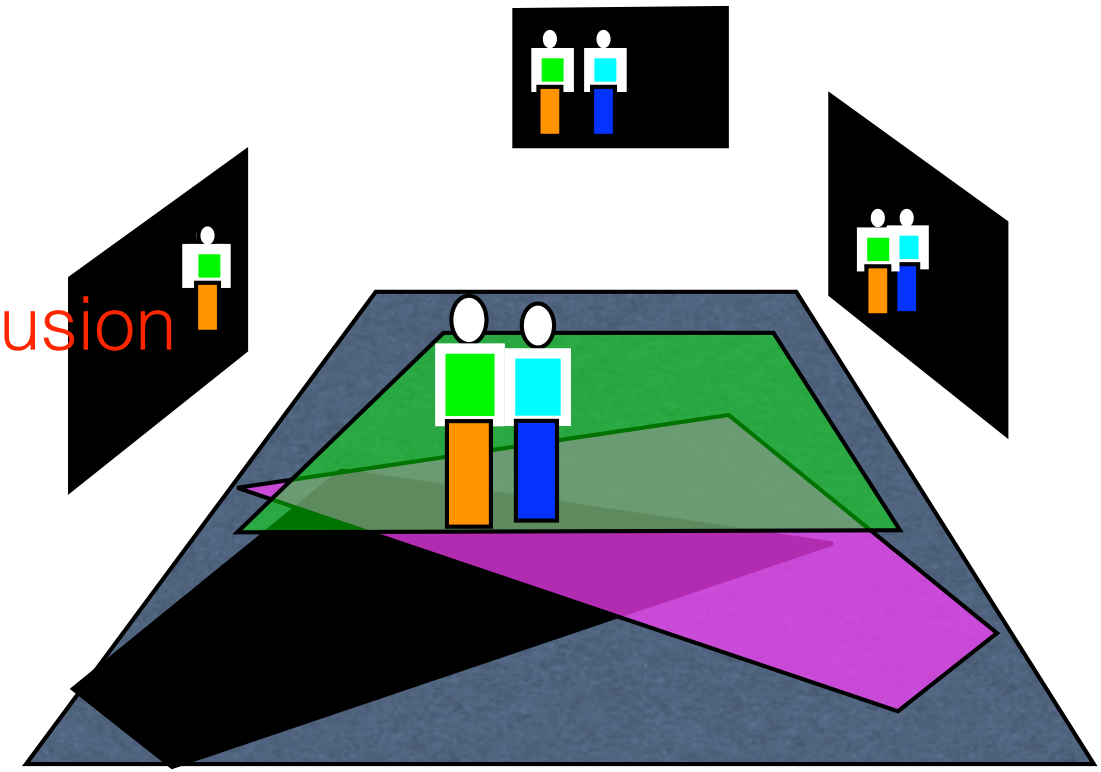
Cameras have a limited field of view

Surveillance of Large Public Spaces



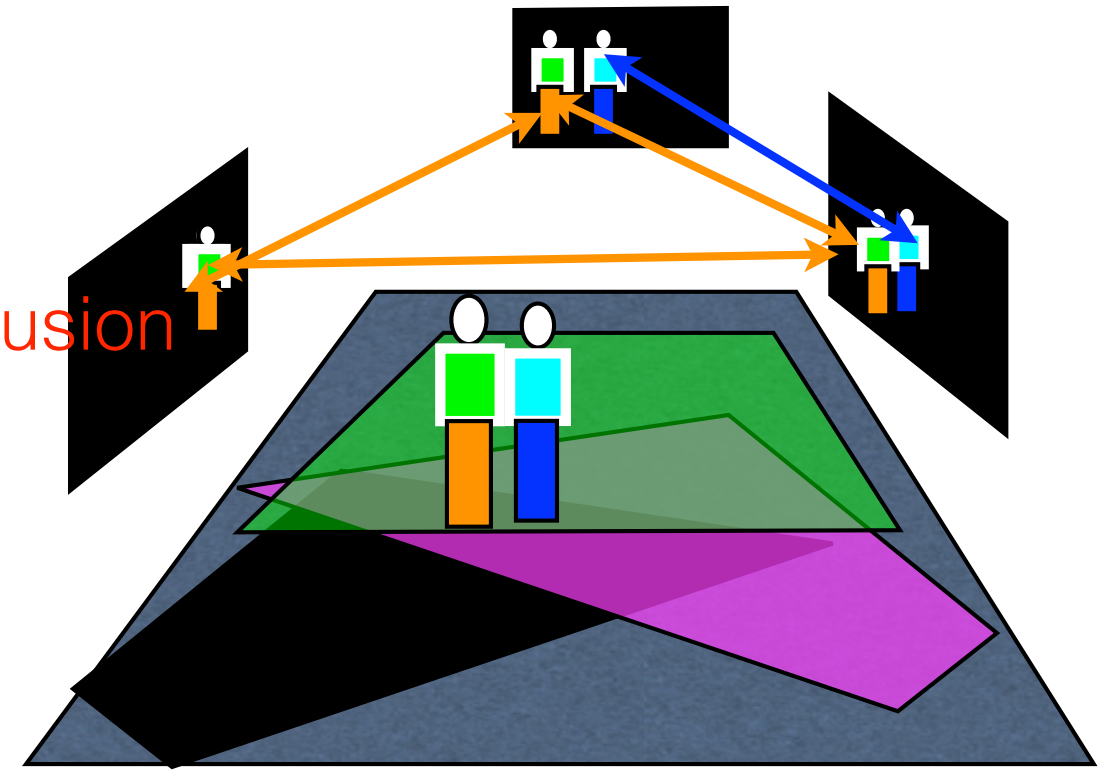
Use multiple cameras to cover a large area.

Surveillance of Large Public Spaces



If a target is occluded in one camera, it is still visible in others.

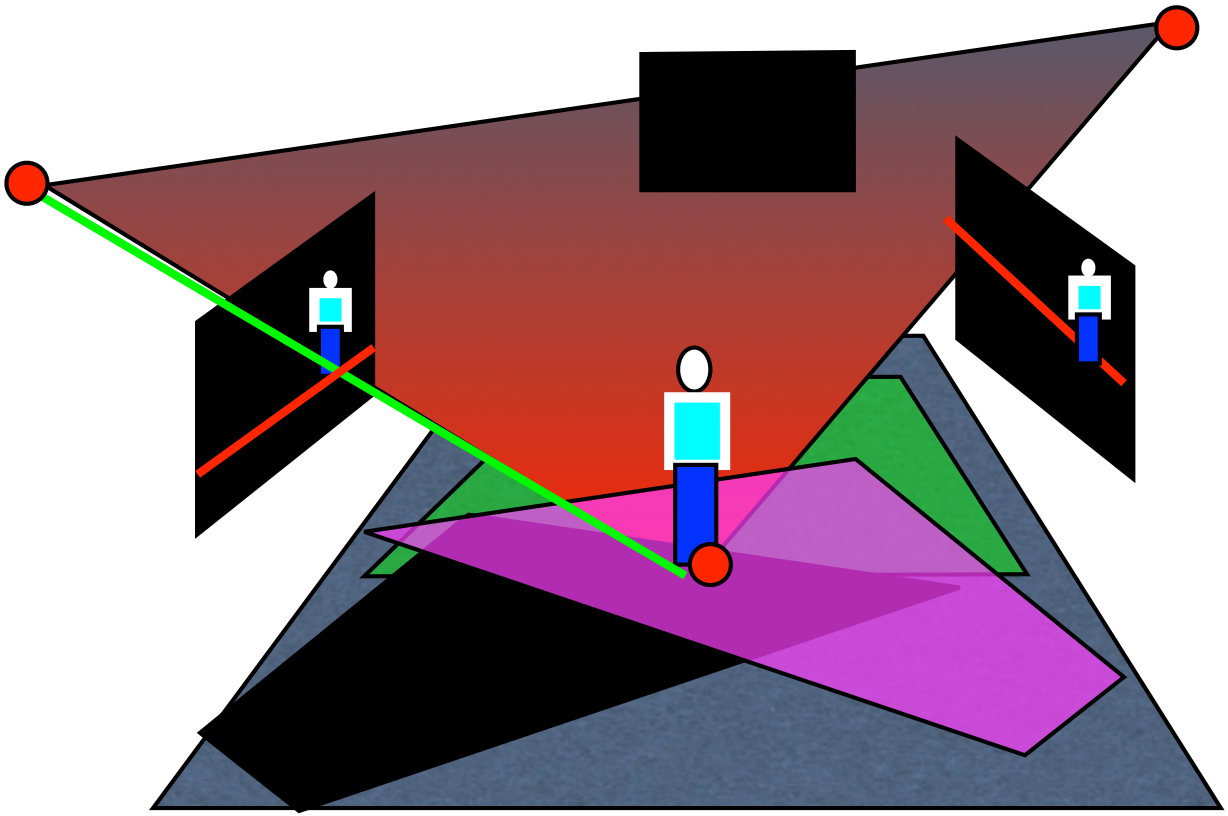
Surveillance of Large Public Spaces



We must solve the correspondence problem.

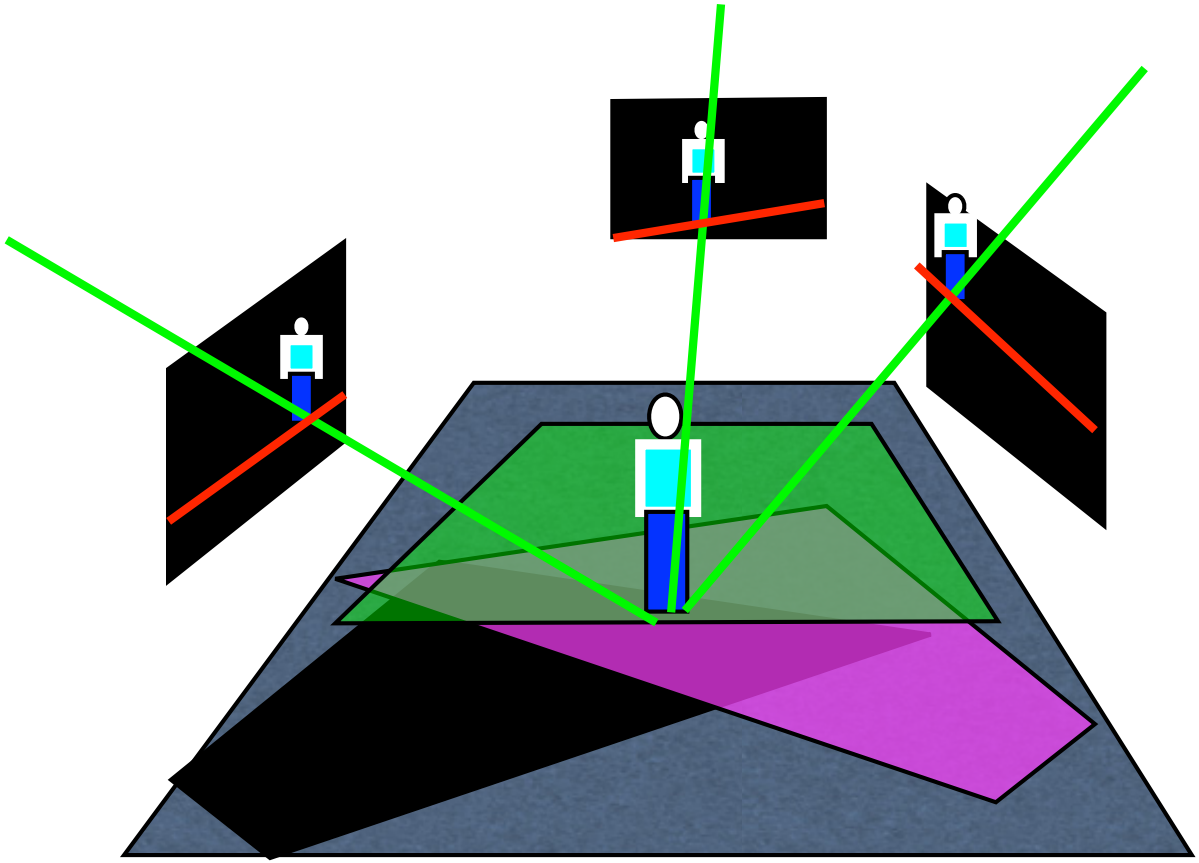
Surveillance of Large Public Spaces

Geometric Constraints: Epipolar Geometry



Surveillance of Large Public Spaces

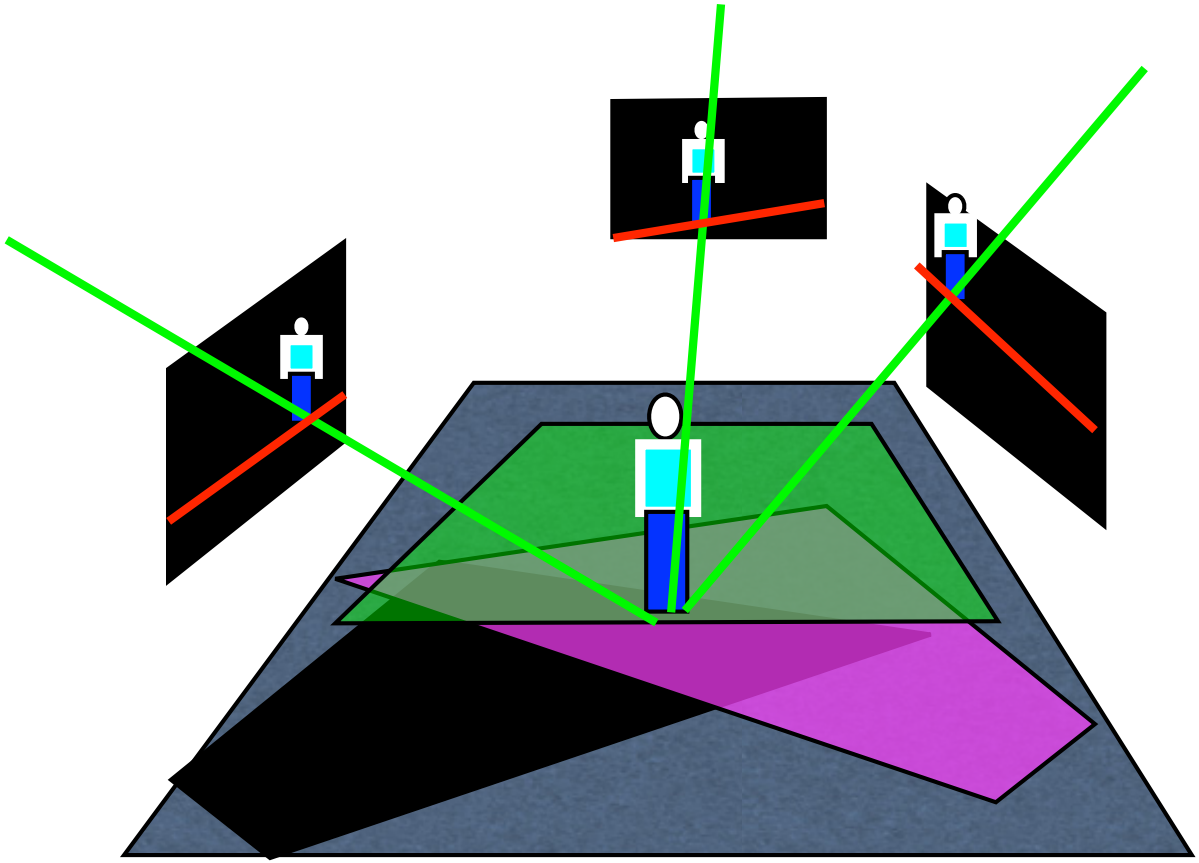
Geometric Constraints: Epipolar Geometry



Corresponding features must lie in corresponding epipolar lines.

Surveillance of Large Public Spaces

3D Assisted Tracker Using Epipolar Geometry:



Corresponding features must lie in corresponding epipolar lines.

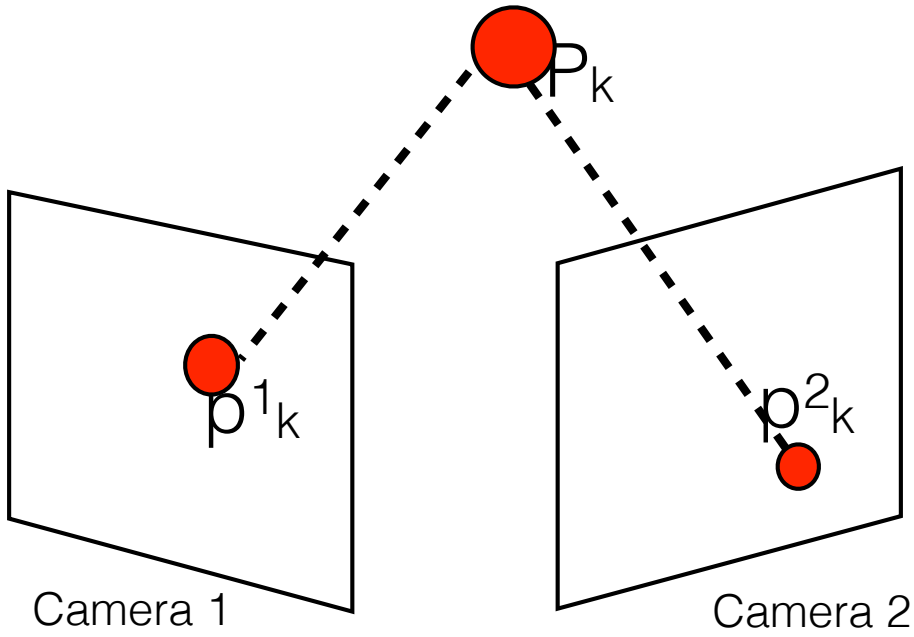
Surveillance of Large Public Spaces

Geometric Constraints: Epipolar Geometry

Tracking failures due to long occlusions

Dynamic-based View Invariant for Multi-camera Tracking

$$p_k^{(1)} = \Pi_1 P_k$$



$$p_k^{(2)} = \Pi_2 P_k$$

Affine projection model:

$$\Pi_j \quad 2 \times 4 \text{ matrix}$$

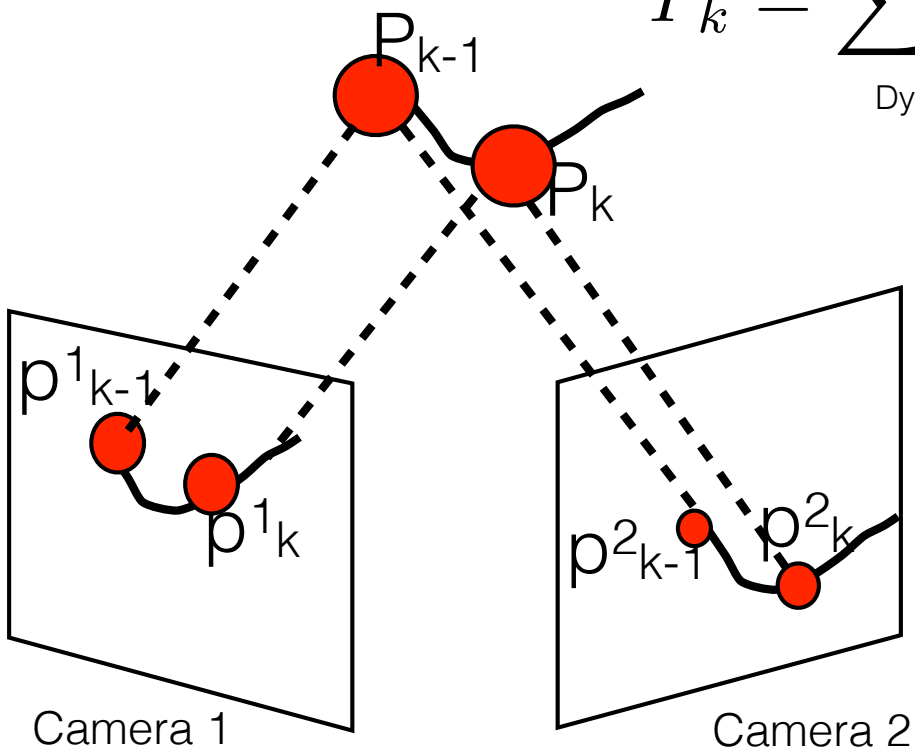
Dynamic-based View Invariant for Multicamera Tracking

$$P_k = \sum a_i P_{k-i}$$

Dynamics Regressor in 3D

$$p_k^{(1)} = \Pi_1 P_k$$

$$p_k^{(2)} = \Pi_2 P_k$$



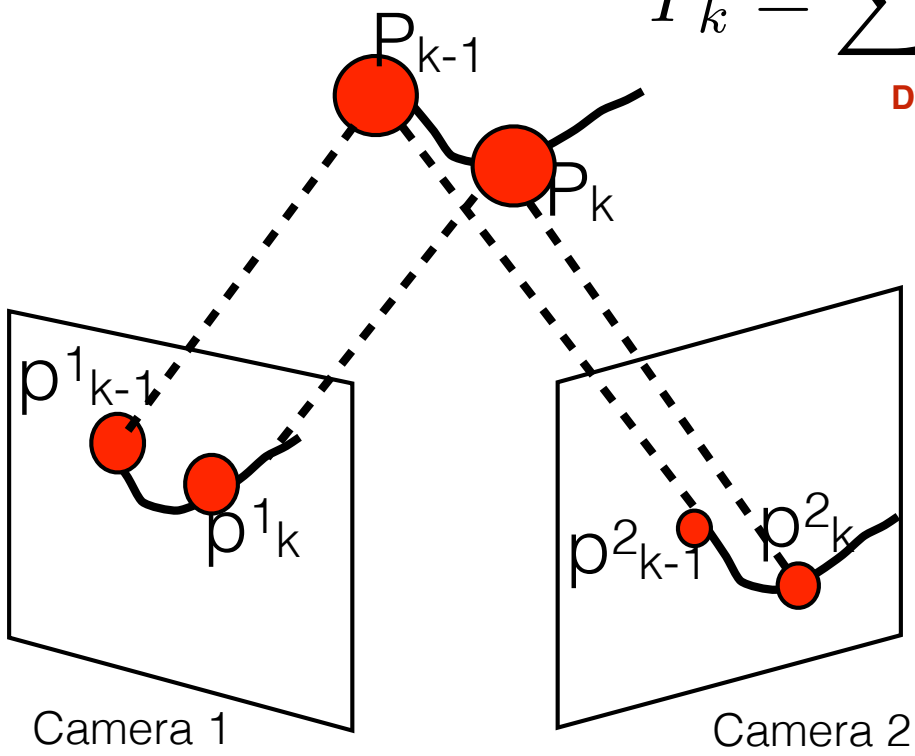
Dynamic-based View Invariant for Multicamera Tracking

$$P_k = \sum a_i P_{k-i}$$

Dynamics Regressor in 3D

$$p_k^{(1)} = \Pi_1 P_k$$

$$p_k^{(2)} = \Pi_2 P_k$$



$$p_k^{(j)} = \Pi_j P_k = \Pi_j \sum a_i P_{k-i} = \sum a_i \Pi_j P_{k-i} = \sum a_i p_{k-i}^{(j)}$$

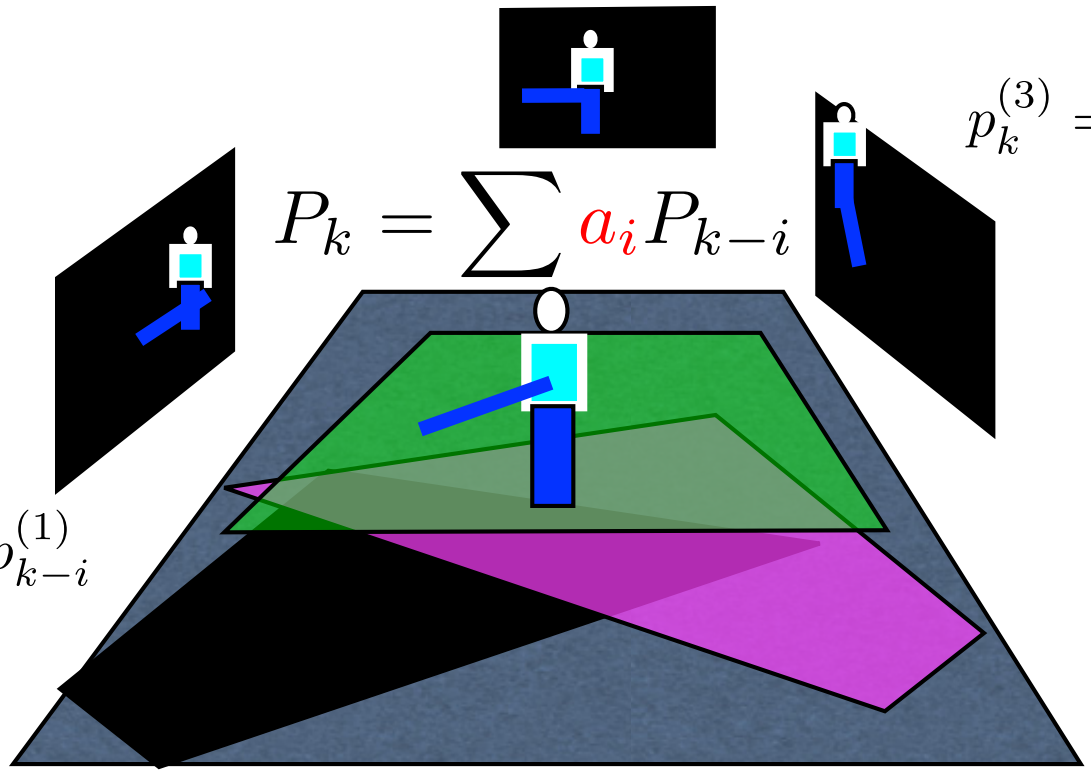
Dynamic Constraints: Trajectories Subspace

$$p_k^{(2)} = \sum a_i p_{k-i}^{(2)}$$

$$p_k^{(3)} = \sum a_i p_{k-i}^{(3)}$$

$$P_k = \sum a_i P_{k-i}$$

$$p_k^{(1)} = \sum a_i p_{k-i}^{(1)}$$



All corresponding trajectories live in the subspace orthogonal to their regressor.

Dynamic-based View Invariant for Multicamera Tracking

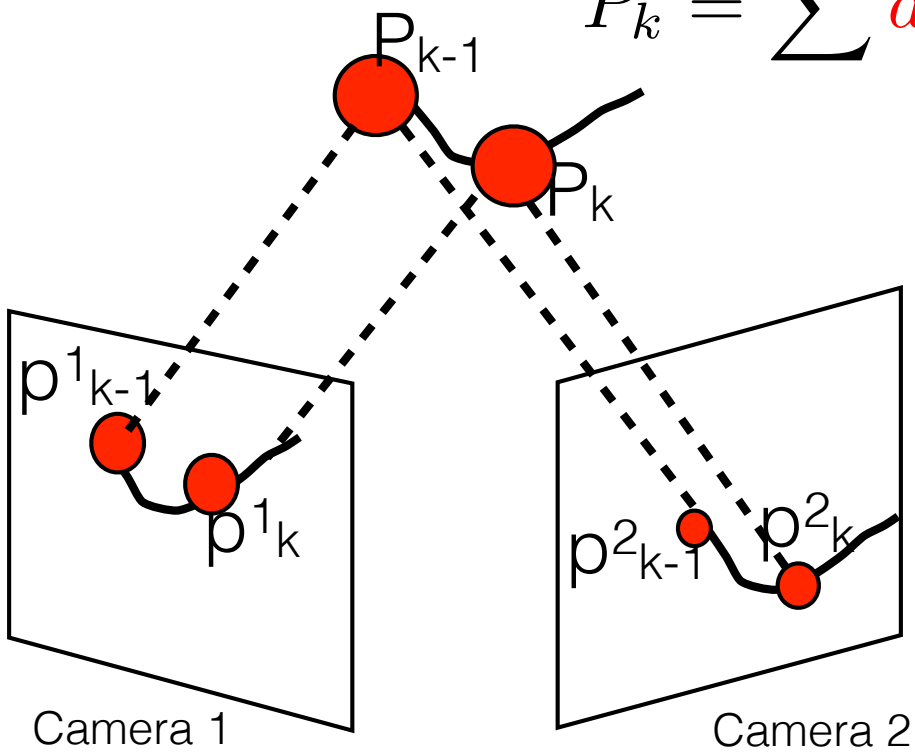
$$P_k = \sum a_i P_{k-i}$$

$$p_k^{(1)} = \Pi_1 P_k$$

$$p_k^{(1)} = \sum a_i p_{k-i}^{(1)}$$

$$p_k^{(2)} = \Pi_2 P_k$$

$$p_k^{(2)} = \sum a_i p_{k-i}^{(2)}$$

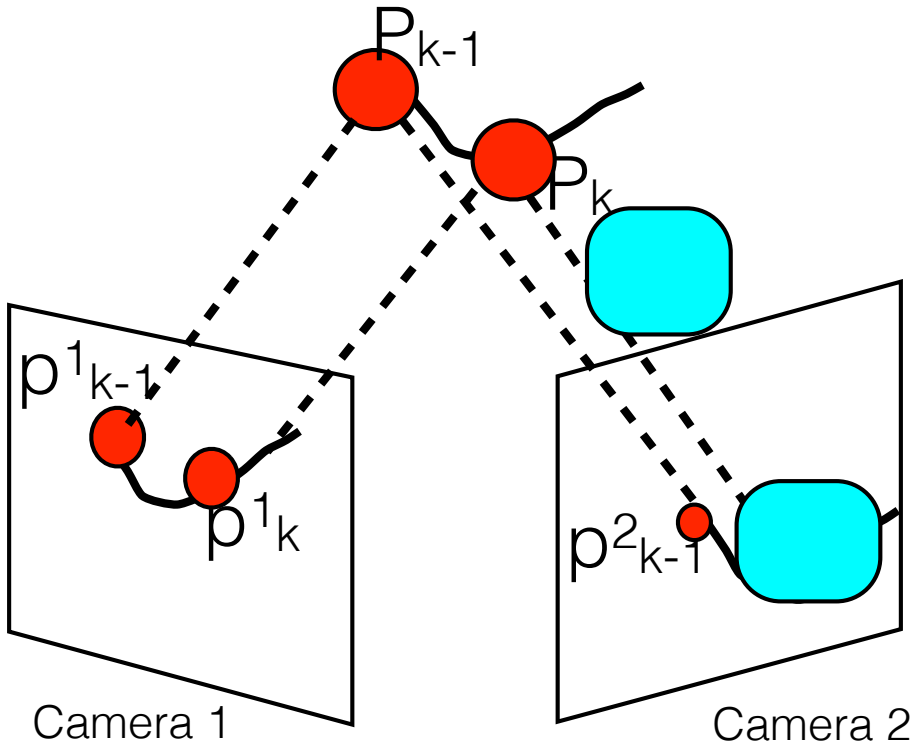


$$\begin{bmatrix} p_k^{(1)} \\ p_k^{(2)} \end{bmatrix} = \begin{bmatrix} p_{k-1}^{(1)} & p_{k-2}^{(1)} & \cdots \\ p_{k-1}^{(2)} & p_{k-2}^{(2)} & \cdots \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

Trajectories' subspaces can be jointly estimated

Dynamic-based View Invariant for Multicamera Tracking

$$p_k^{(1)} = \sum a_i p_{k-i}^{(1)}$$



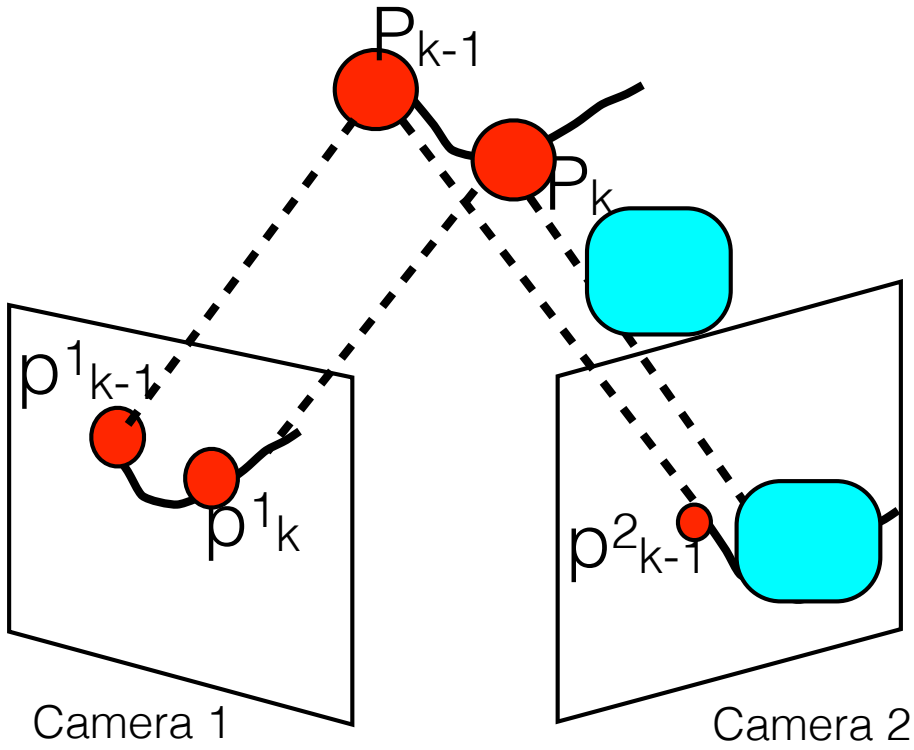
$$p_k^{(2)} = \sum a_i p_{k-i}^{(2)}$$

$$\begin{bmatrix} p_k^{(1)} \\ p_k^{(2)} \end{bmatrix} = \begin{bmatrix} p_{k-1}^{(1)} & p_{k-2}^{(1)} & \cdots \\ p_{k-1}^{(2)} & p_{k-2}^{(2)} & \cdots \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

If Camera 2 is occluded, Camera 1 takes over

Dynamic-based View Invariant for Multicamera Tracking

$$p_k^{(1)} = \sum a_i p_{k-i}^{(1)}$$

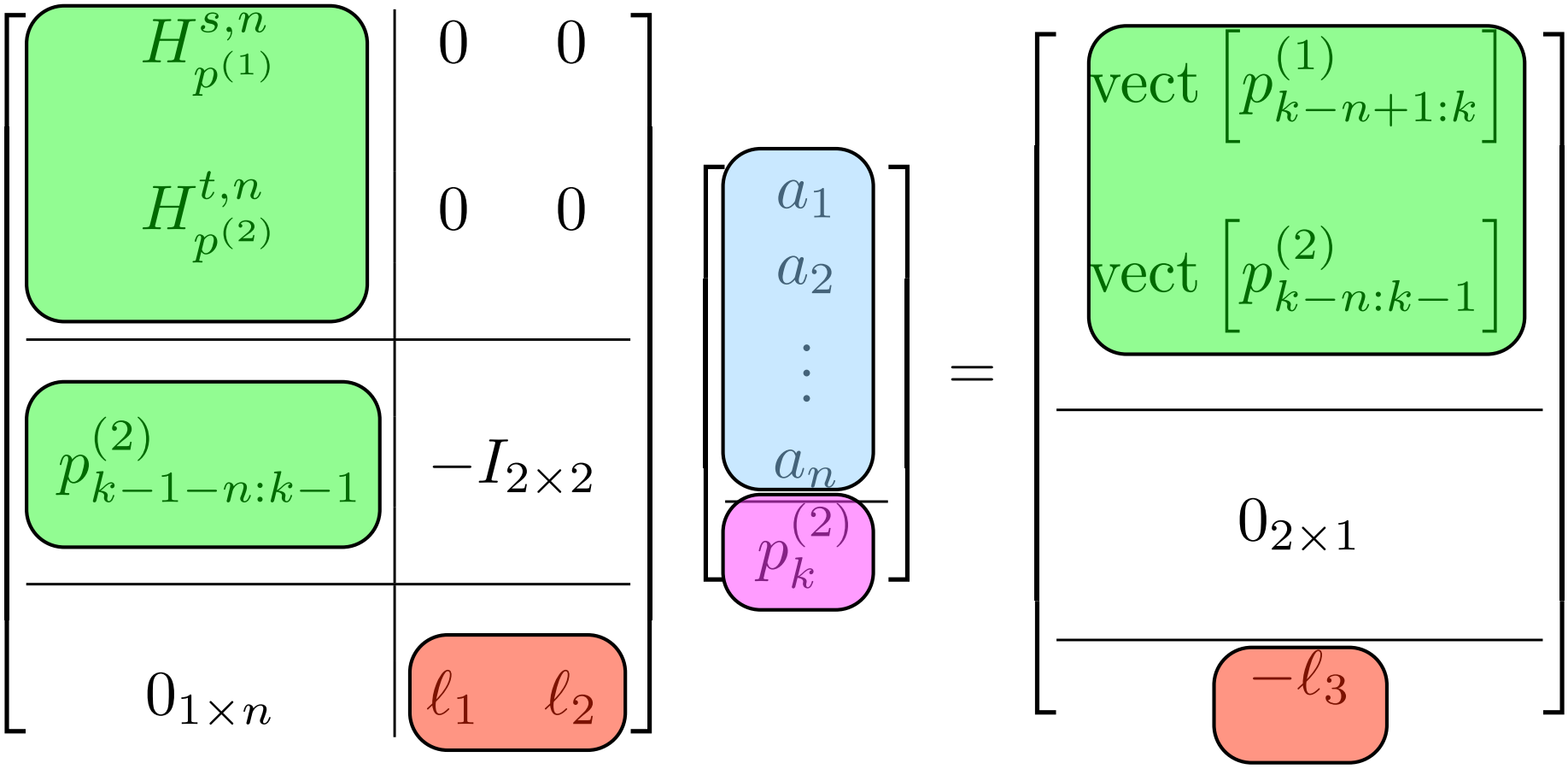


$$p_k^{(2)} = \sum a_i p_{k-i}^{(2)}$$

$$\begin{bmatrix} p_k^{(1)} \\ (2) \end{bmatrix} = \begin{bmatrix} p_{k-1}^{(1)} & p_{k-2}^{(1)} & \cdots \\ (2) & (2) & \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

If Camera 2 is occluded, Camera 1 takes over

Jointly use of geometry and dynamics



where

$$\begin{bmatrix} p_k^{(1)^T} & 1 \end{bmatrix} F = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}$$

Surveillance of Large Public Spaces

Dynamic Constraints: Trajectories Subspace

If a target is occluded in one camera, it is still be visible in others.

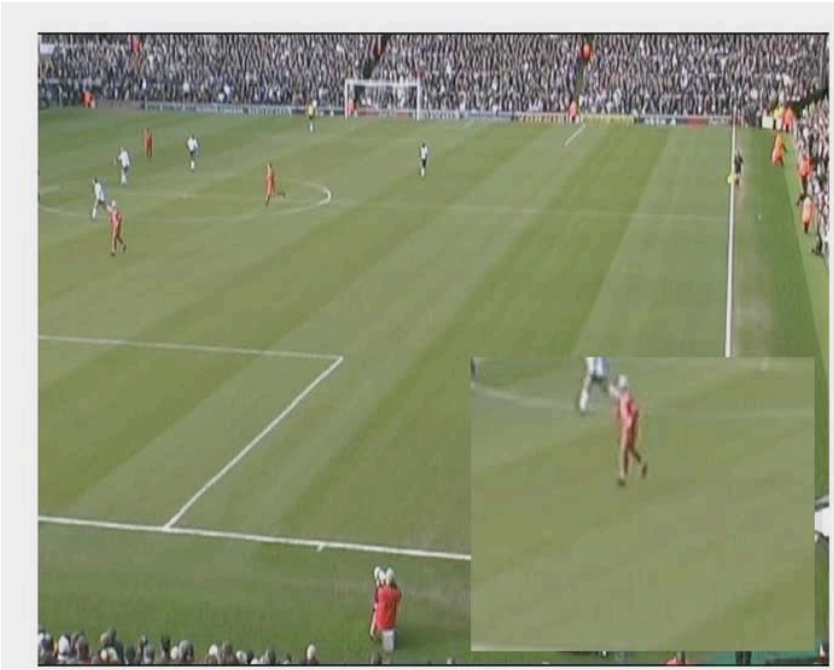
Surveillance of Large Public Spaces

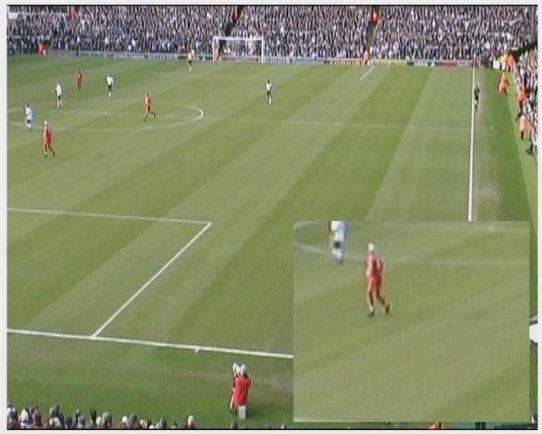
Dynamic Constraints: Trajectories Subspace

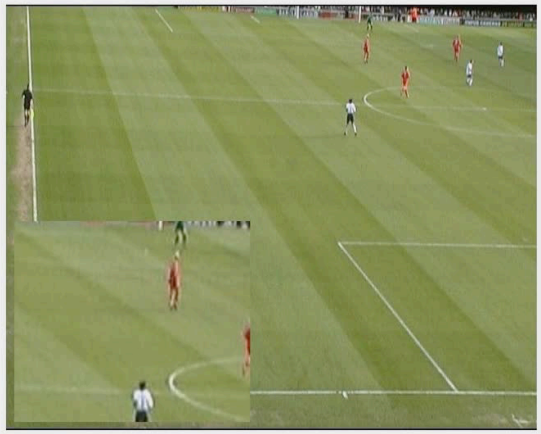
If a target is occluded in one camera, it is still be visible in others.

Multi-camera Tracking

Our Method







Without using dynamic constraints

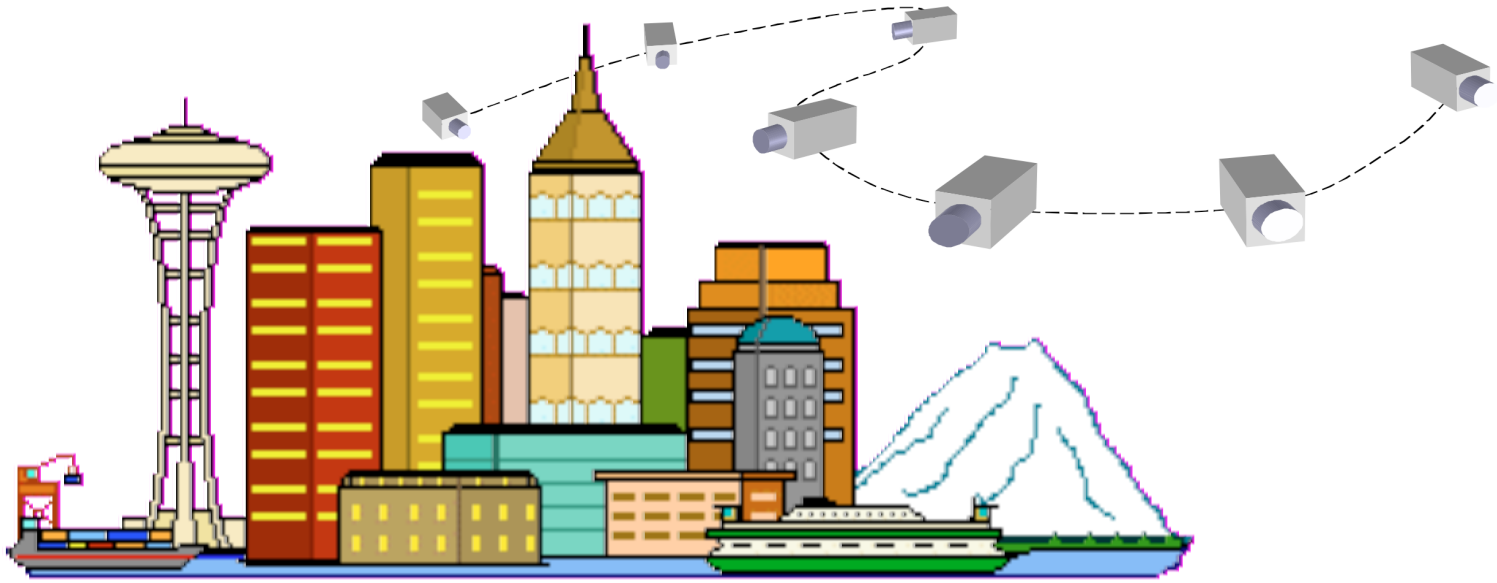
Affine Structure from Motion by Factorization

Tomasi and Kanade, 1992

Structure from Motion

Given a set of flow fields or displacement vectors from a moving camera over time, determine:

- The sequence of camera poses
- The 3D structure of the scene



Killer Application

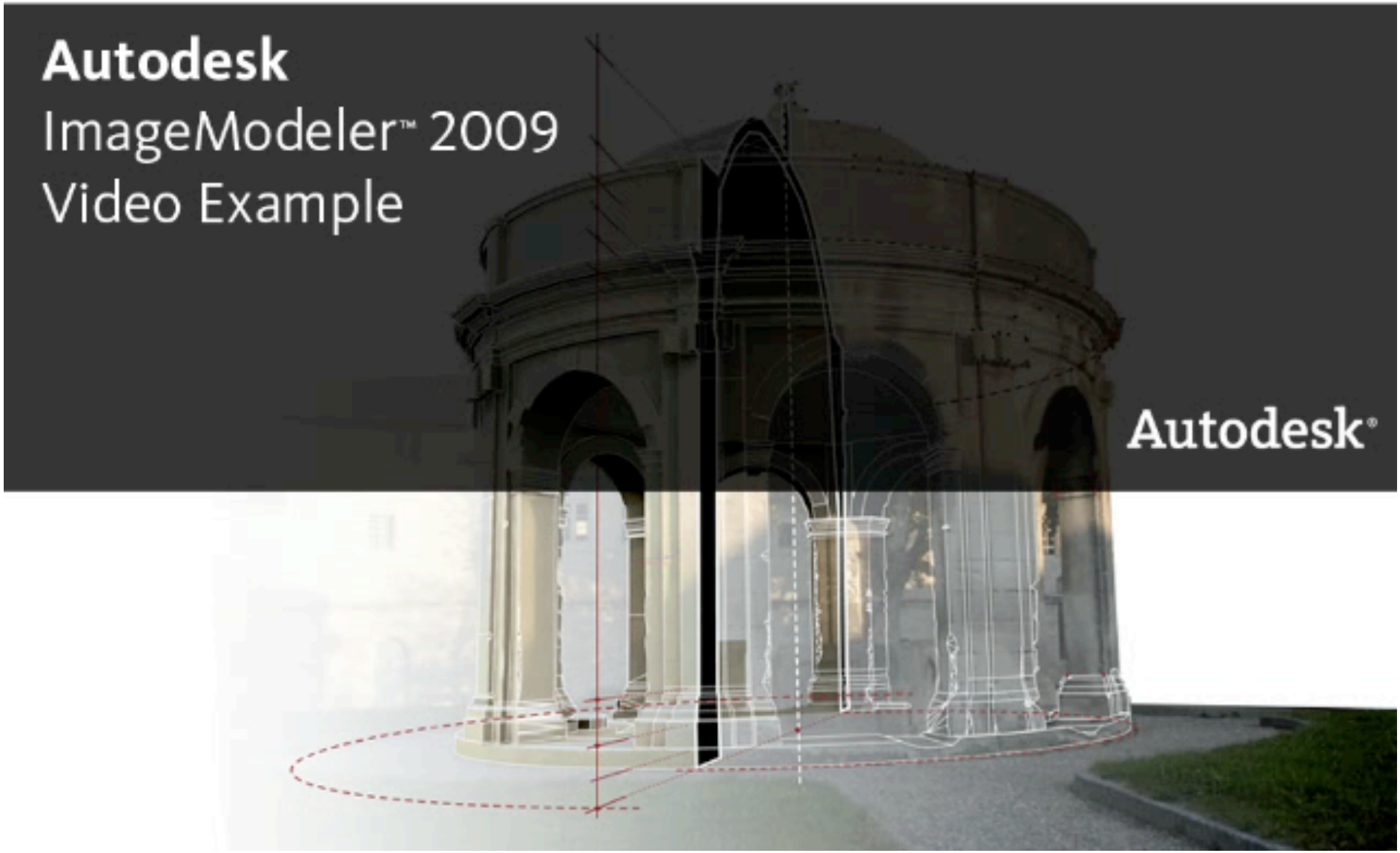
Image Modeler:

Track a set of feature point through a movie sequence

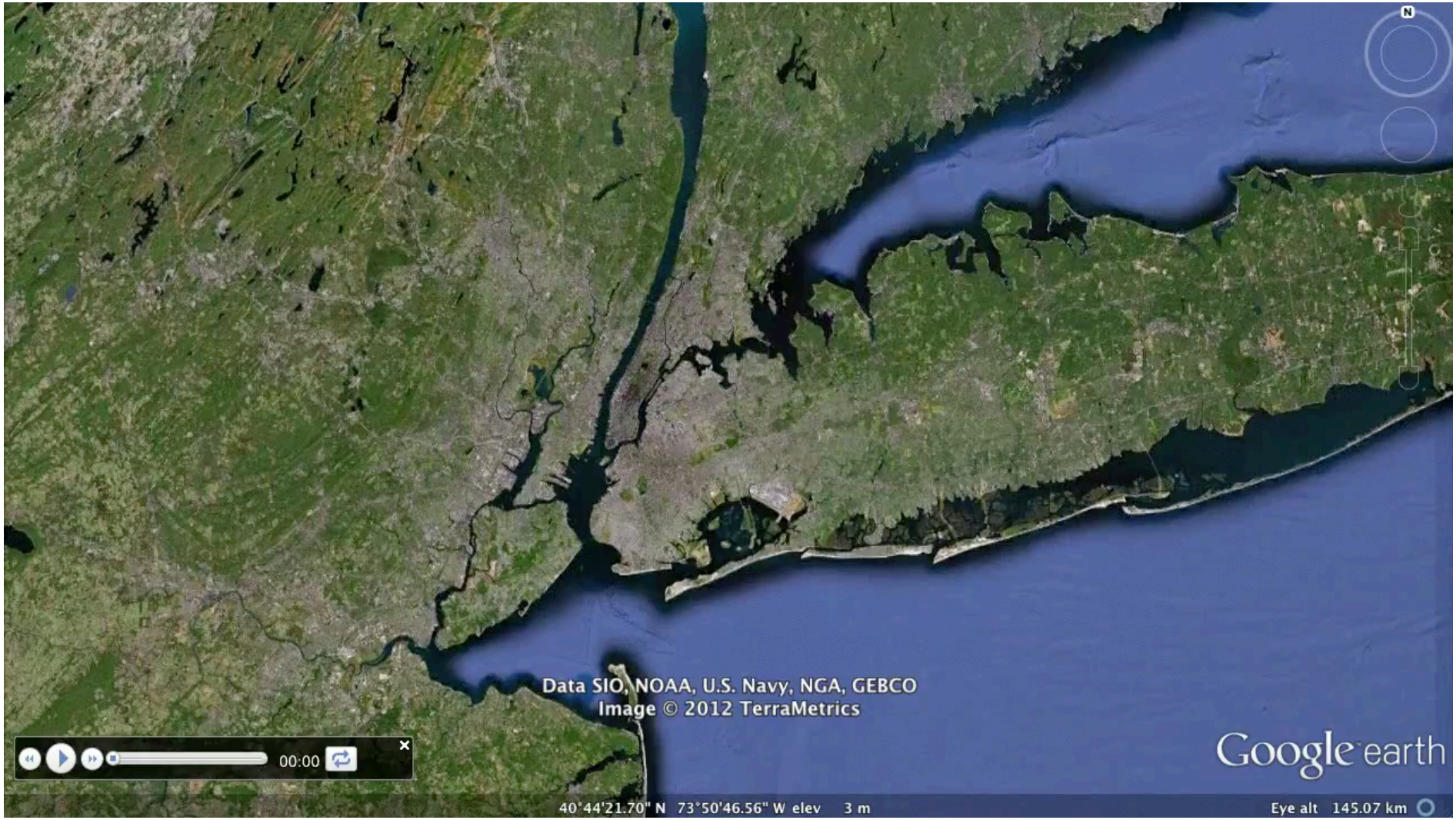
Deduce where the cameras are and the 3D locations of the points that were tracked

Render synthetic objects with respect to the deduced 3D geometry of the scene/cameras.

http://area.autodesk.com/showcase/movies/salzburg_museum_modeling_and_texturing_with_image modeler


Autodesk
ImageModeler™ 2009
Video Example
Autodesk®

J. Xiao & T. Furukawa (ECCV 2012)



SFM by Factorization

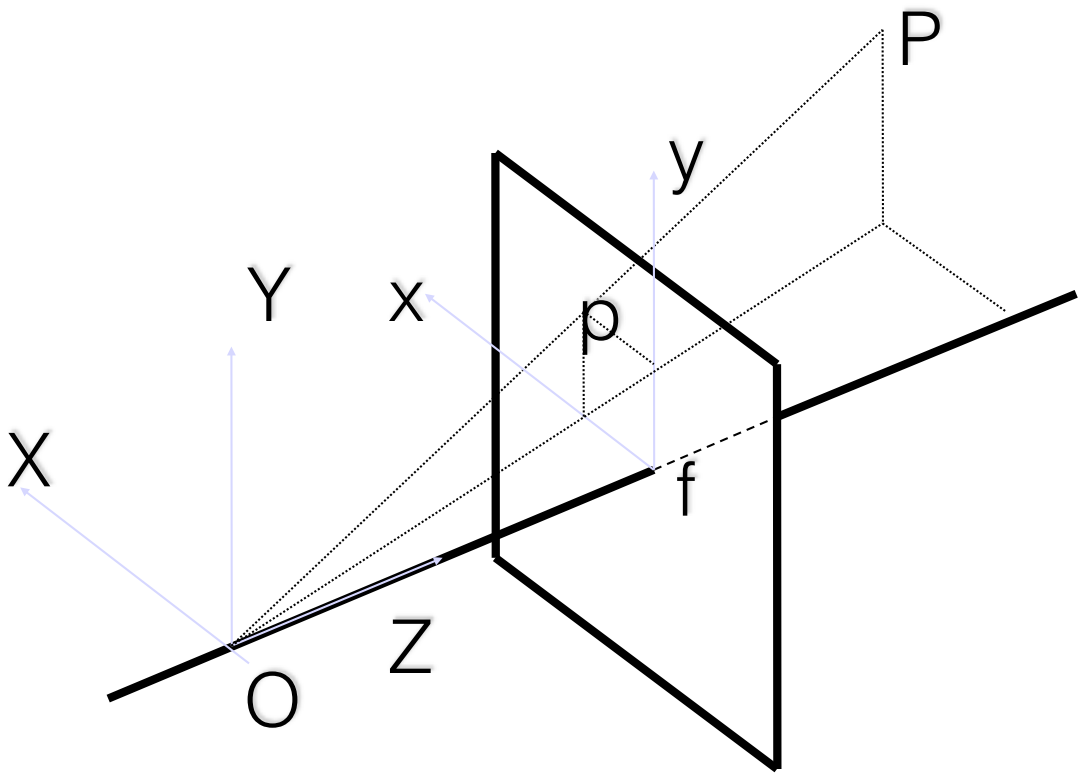
Consider a FIXED camera and a MOVING object

Track n features across m frames.

We want to find the structure and the motion of the object.

Pinhole Camera Model

(Camera Coordinates)

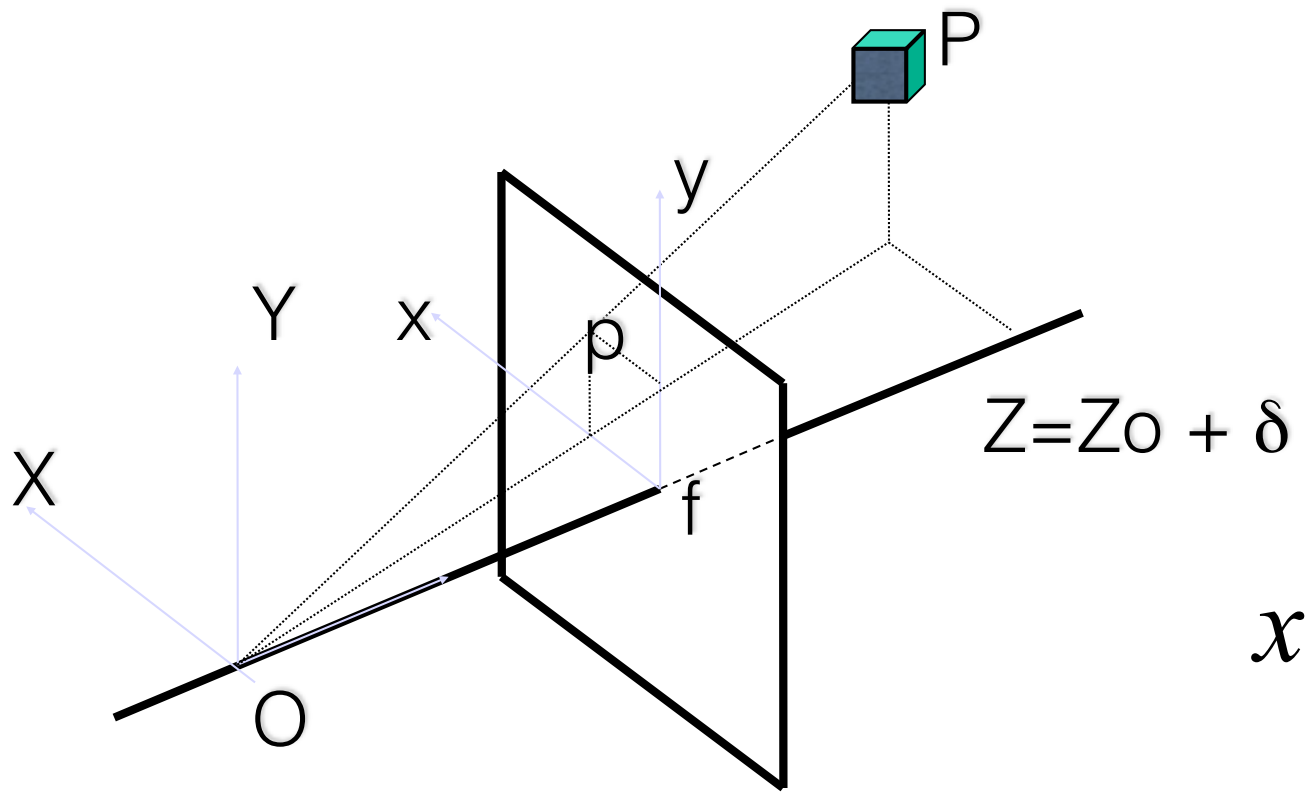


$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

- X,Y,Z are expressed in the camera coordinate system

Weak Perspective Model



$$x \approx f \frac{X}{Z_o}$$

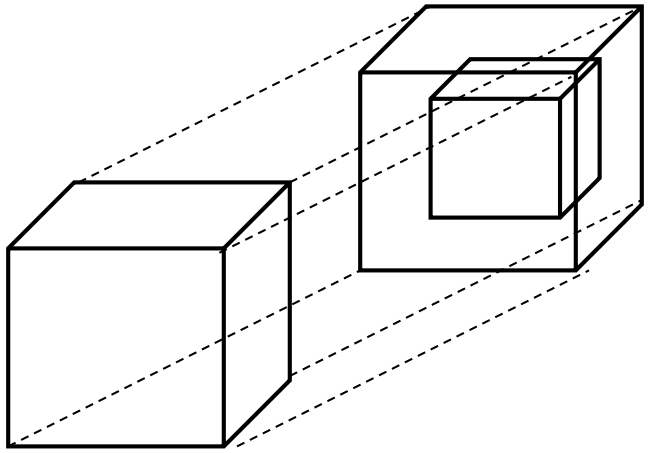
$$y \approx f \frac{Y}{Z_o}$$

- Object depth δ << Camera distance Zo
- Linear equations !!

Weak Perspective Model

$$x \approx f \frac{X}{Z_o}$$

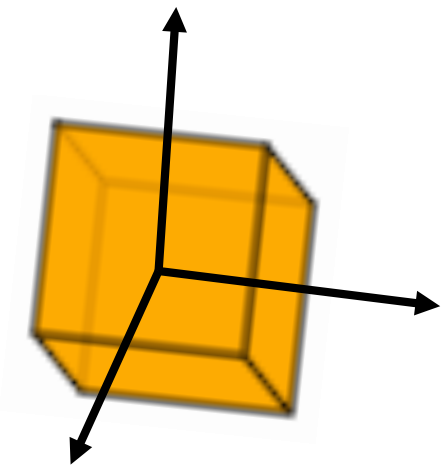
$$y \approx f \frac{Y}{Z_o}$$



Weak perspective = Orthographic projection +
Isotropic Scaling

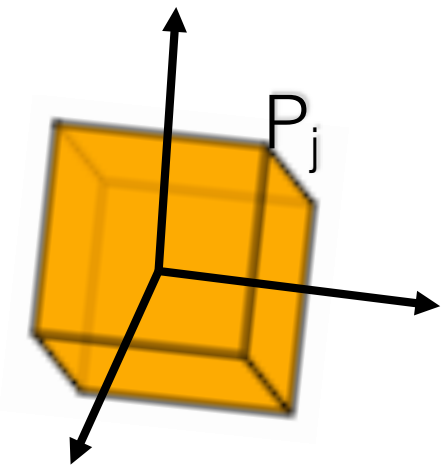
Using Object Coordinates:

Using Object Coordinates:



1. Attach a coordinate system to the object
(it moves WITH the object)

Using Object Coordinates:

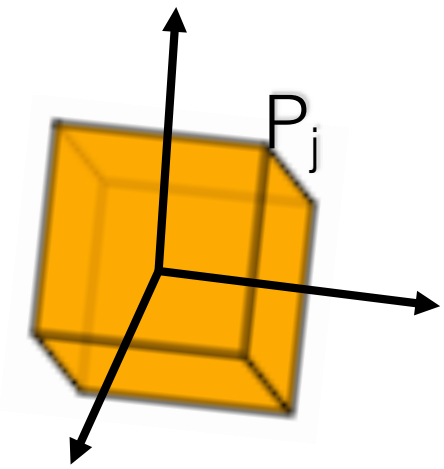


1. Attach a coordinate system to the object (it moves WITH the object)

2. Apply a transformation (R&T) to express object coordinates in the camera coordinate system.

NOTE: Since the object is moving with respect to the camera, the transformation to change coordinate systems changes from frame to frame

Using Object Coordinates:



1. Attach a coordinate system to the object (it moves WITH the object)

2. Apply a transformation (R&T) to express object coordinates in the camera coordinate system.

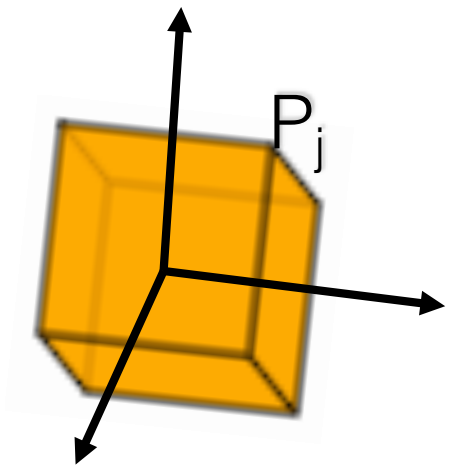
3. Apply weak perspective model: scale first two coordinates and drop third coordinate

Using Object Coordinates:

Affine projection of point Pj at frame i:

$$p_{ij} = A_i P_j + b_i$$

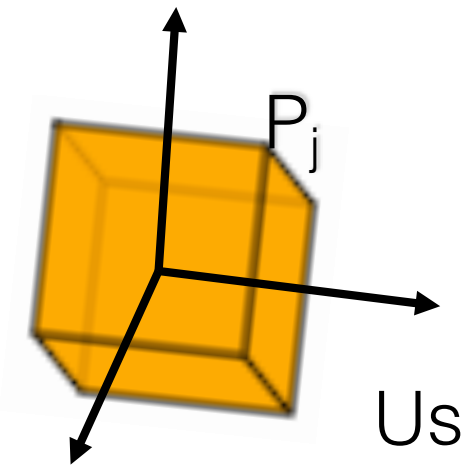
2x1 2x3 3x1 2x1



Using Object Coordinates:

Affine projection of point Pj at frame i:

$$\underset{2x1}{p_{ij}} = \underset{2x3}{A_i}\underset{3x1}{P_j} + \underset{2x1}{b_i}$$

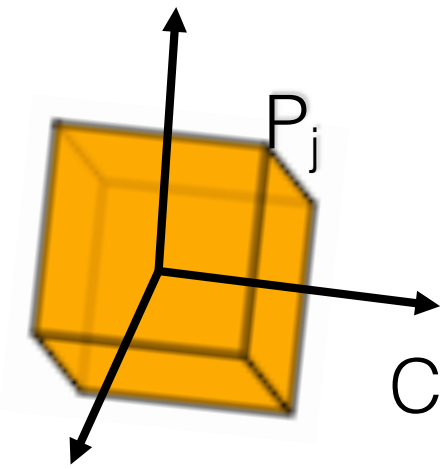


Using homogeneous coordinates:

$$\underset{2x1}{p_{ij}} = \underset{2x4}{\begin{bmatrix} A_i & b_i \end{bmatrix}} \begin{bmatrix} P_j \\ 1 \end{bmatrix}_{4x1}$$

Using Object Coordinates:

Affine projection of point Pj at frame i:



$$p_{ij} = \begin{bmatrix} A_i & b_i \end{bmatrix} \begin{bmatrix} P_j \\ 1 \end{bmatrix}$$

p_{ij}: 2x1; $\begin{bmatrix} A_i & b_i \end{bmatrix}$: 2x4; $\begin{bmatrix} P_j \\ 1 \end{bmatrix}$: 4x1

Considering all points Pj j=1,…,n:

$$\begin{bmatrix} p_{i1} & p_{i2} & \dots & p_{in} \end{bmatrix} = \begin{bmatrix} A_i & b_i \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \dots & P_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

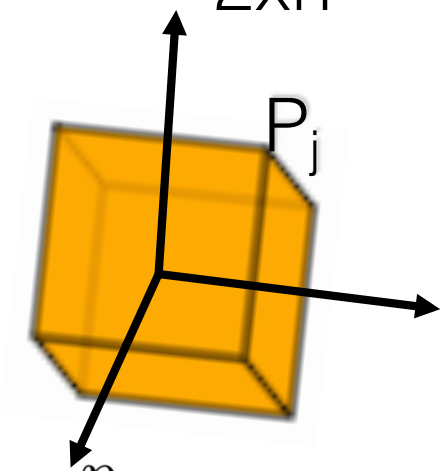
$\begin{bmatrix} p_{i1} & p_{i2} & \dots & p_{in} \end{bmatrix}$: 2xn; $\begin{bmatrix} A_i & b_i \end{bmatrix}$: 2x4; $\begin{bmatrix} P_1 & P_2 & \dots & P_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$: 4xn

Using Object Coordinates:

Projections at frame i:

$$\begin{bmatrix} p_{i1} & p_{i2} & \dots & p_{in} \end{bmatrix} = \begin{bmatrix} A_i & b_i \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \dots & P_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

2xn 2x4 4xn

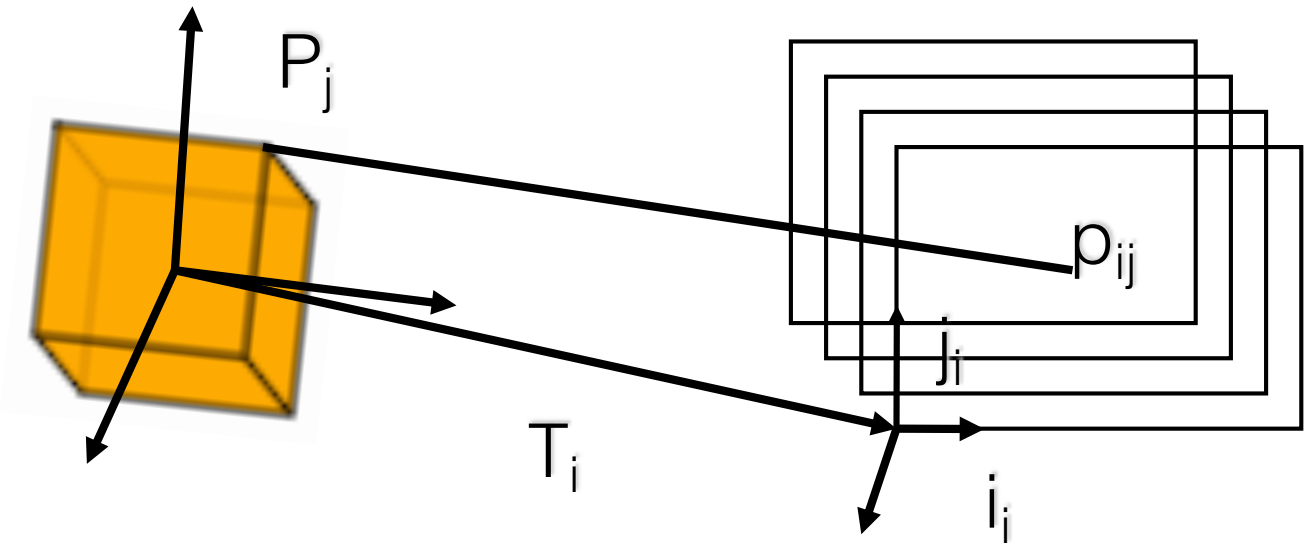


Collecting all frames i=1,…,m

$$\begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \dots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{mn} \end{bmatrix} = \begin{bmatrix} A_1 & b_1 \\ A_2 & b_2 \\ \vdots & \vdots \\ A_m & b_m \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \dots & P_n \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

2mxn 2mx4 4xn

$$p_{ij} = \begin{bmatrix} A_i & b_i \end{bmatrix} \begin{bmatrix} P_j \\ 1 \end{bmatrix}$$



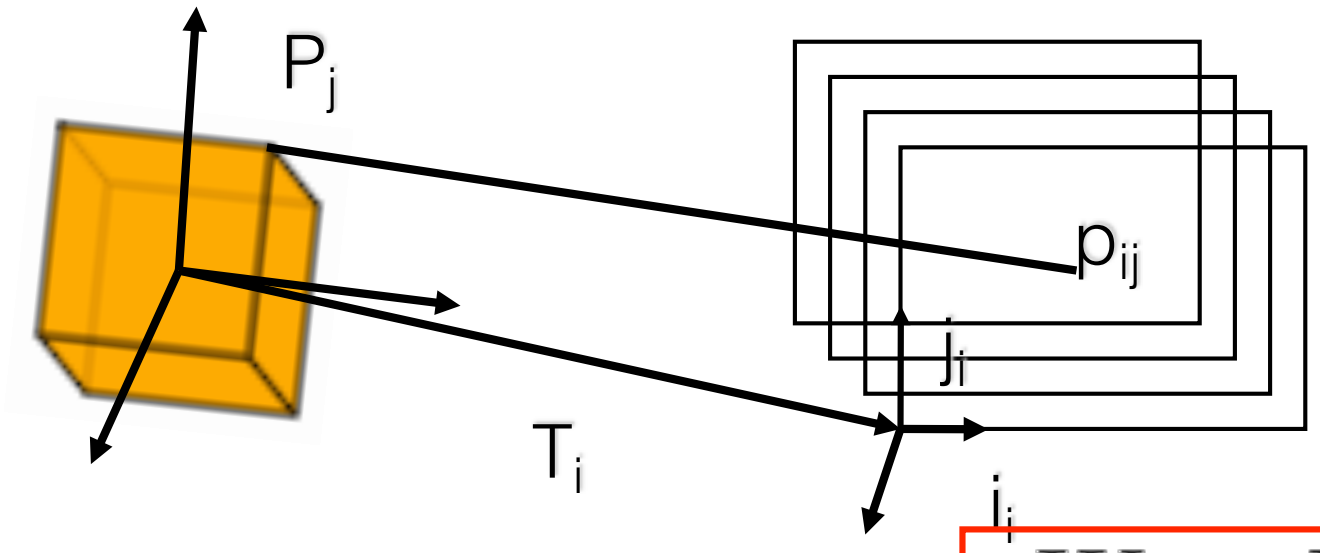
measurements

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & W & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix} = \begin{bmatrix} A_1 & b_1 \\ A_2 & b_2 \\ \vdots & M \\ A_m & b_m \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \cdots & P_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} S$$

motion

structure

$$W = MS$$



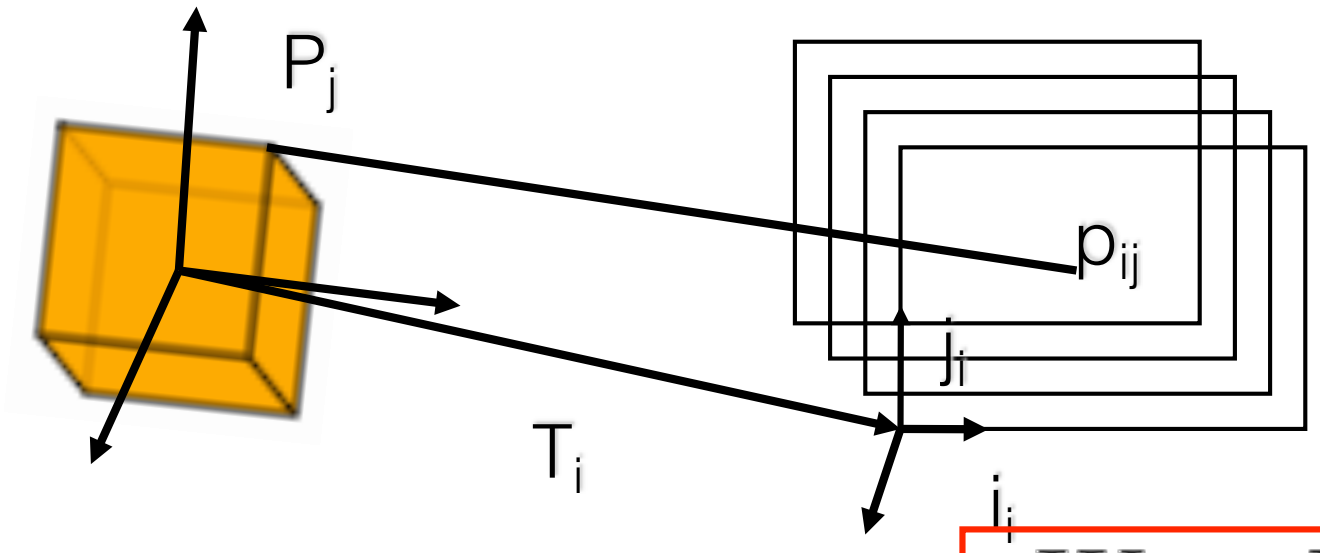
$$W = MS$$

2m x n 2m x 4 4 x n

W is rank 4 and M and S can be found (up to an affine transf.) from the SVD of W

$$W = UDV^T$$

$$M = UD^{\frac{1}{2}}Q \qquad S = Q^{-1}D^{\frac{1}{2}}V^T$$



$$W = MS$$

2m x n 2m x 4 4 x n

W is rank 4 and M and S can be found (up to an affine transf.) from the SVD of W

$$W = UDV^T$$

In general, W will be full rank (due to noise) but we can force it to be rank 4 by zeroing the singular values 5 to 2m

W would be closer to rank 4 if …

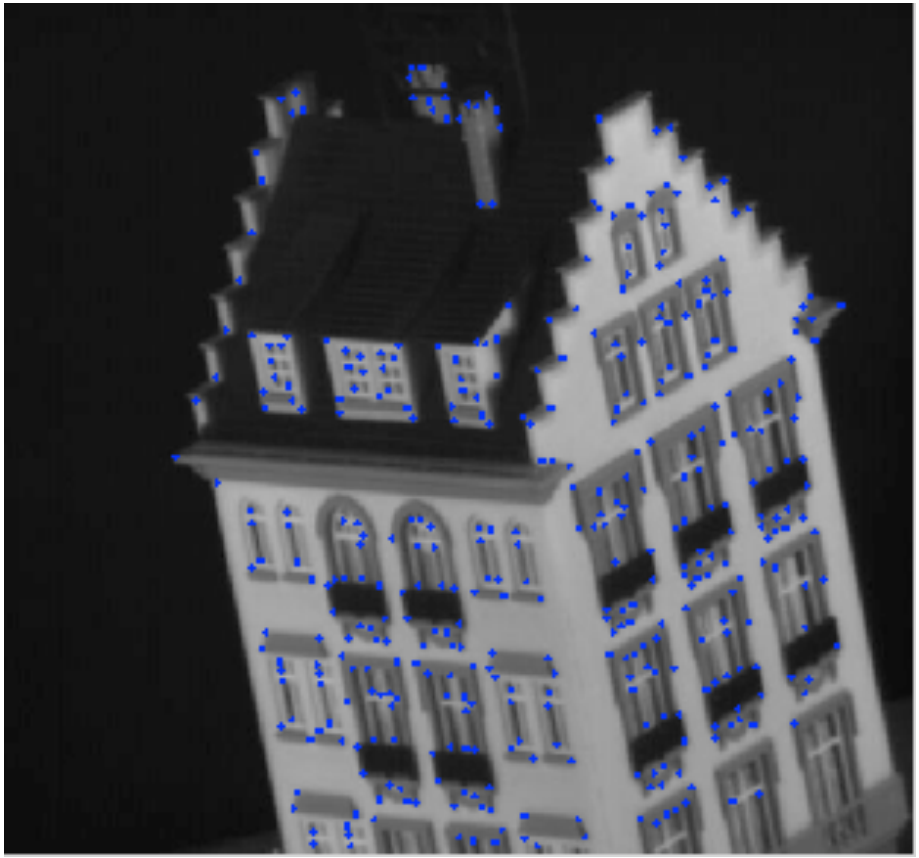
Coordinates of tracked features are more accurate.

Structure from Dynamics

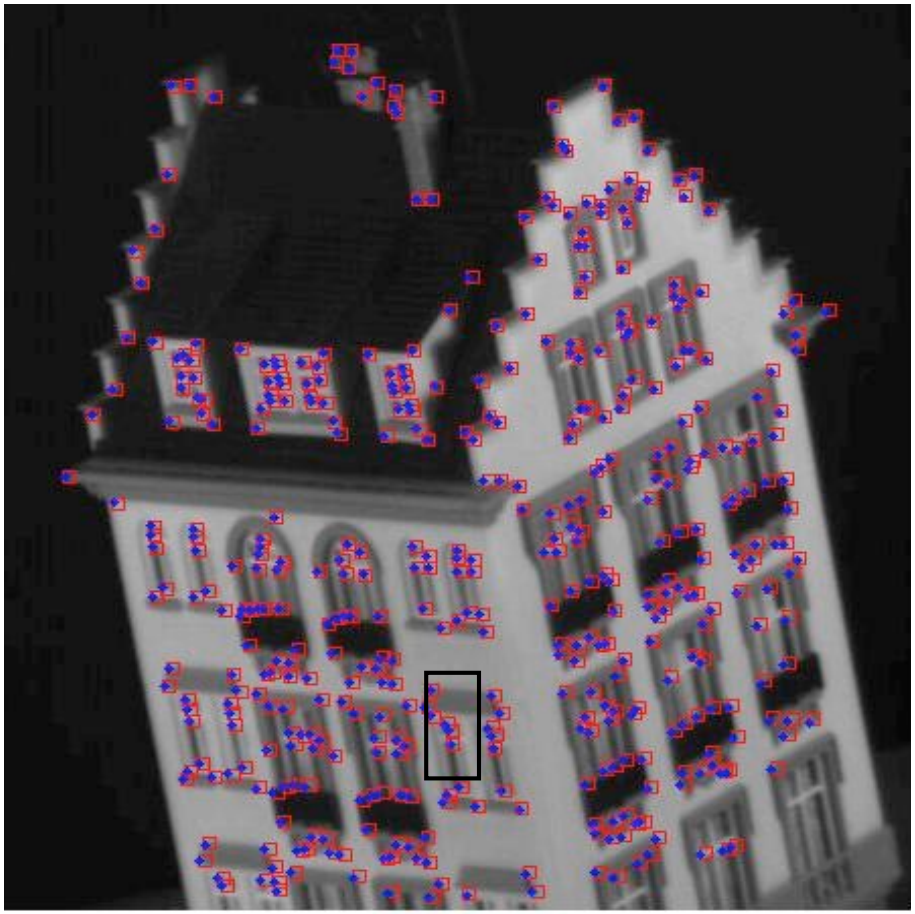
Fransen, Camps, Sznaier

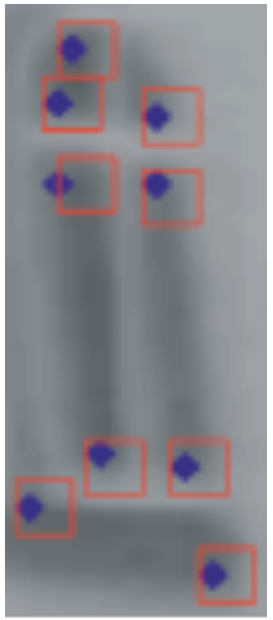
ICCV 05

House Sequence



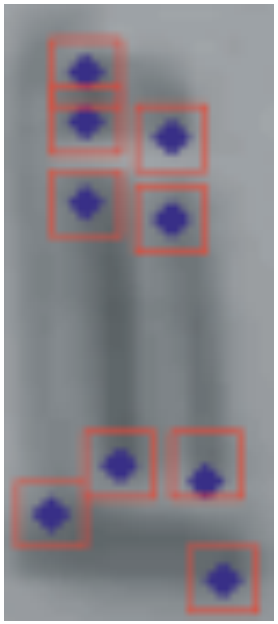
Bias and Search Window



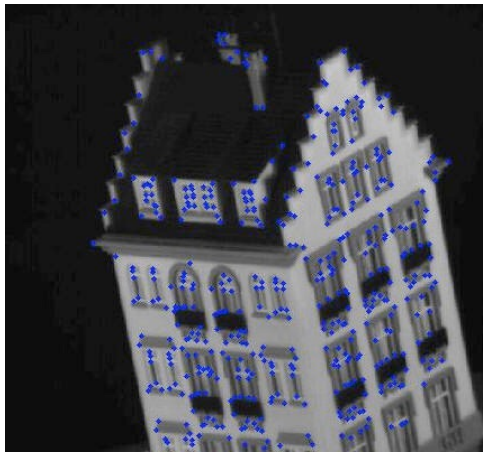


Without Motion

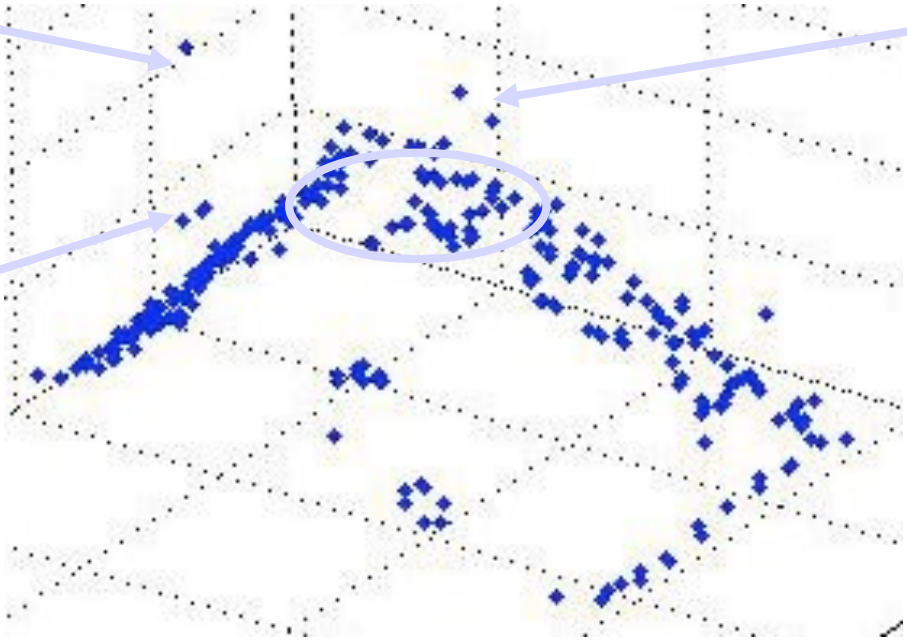
With CF Identified Motion



SfM Factorization


outliers



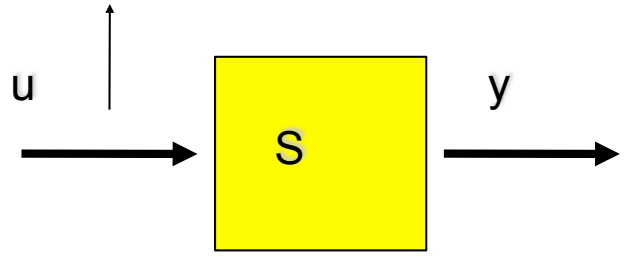
Observations

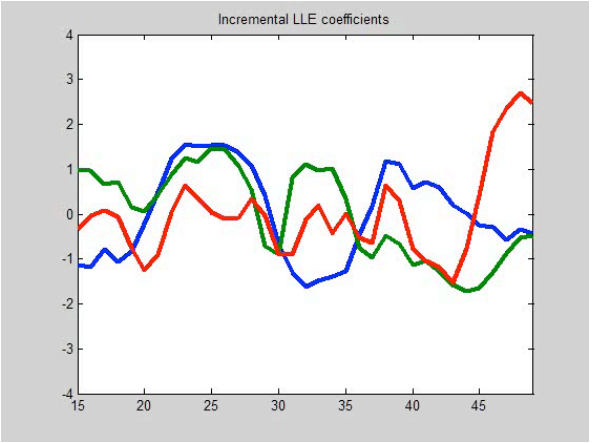
Swapping rows in W does not affect the results obtained by factorization.

The method exploits geometrical constraints (the object is rigid) but does not exploit temporal information – i.e. the order of the frames.

Temporal information can be incorporated by modeling the dynamics of the motion

Looking for the Hidden Dynamics





We can model temporally evolving data as the output of *unknown* dynamic systems.

- Data prediction
- Recovery of missing data
- Event detection
- Temporal correlations
- Often, no need to identify the systems !

SfM Factorization with Dynamics

Find R, T and S from a few frames

Use dynamics to:

- Predict future R and T
- Use predictions to get better correspondences

Refine R, T and S

Capturing Dynamics from Experimental Data: The Hankel Matrix

Given a sequence of measurements of d-dimensional vectors:

$$y_1, y_2, y_3 \cdots$$

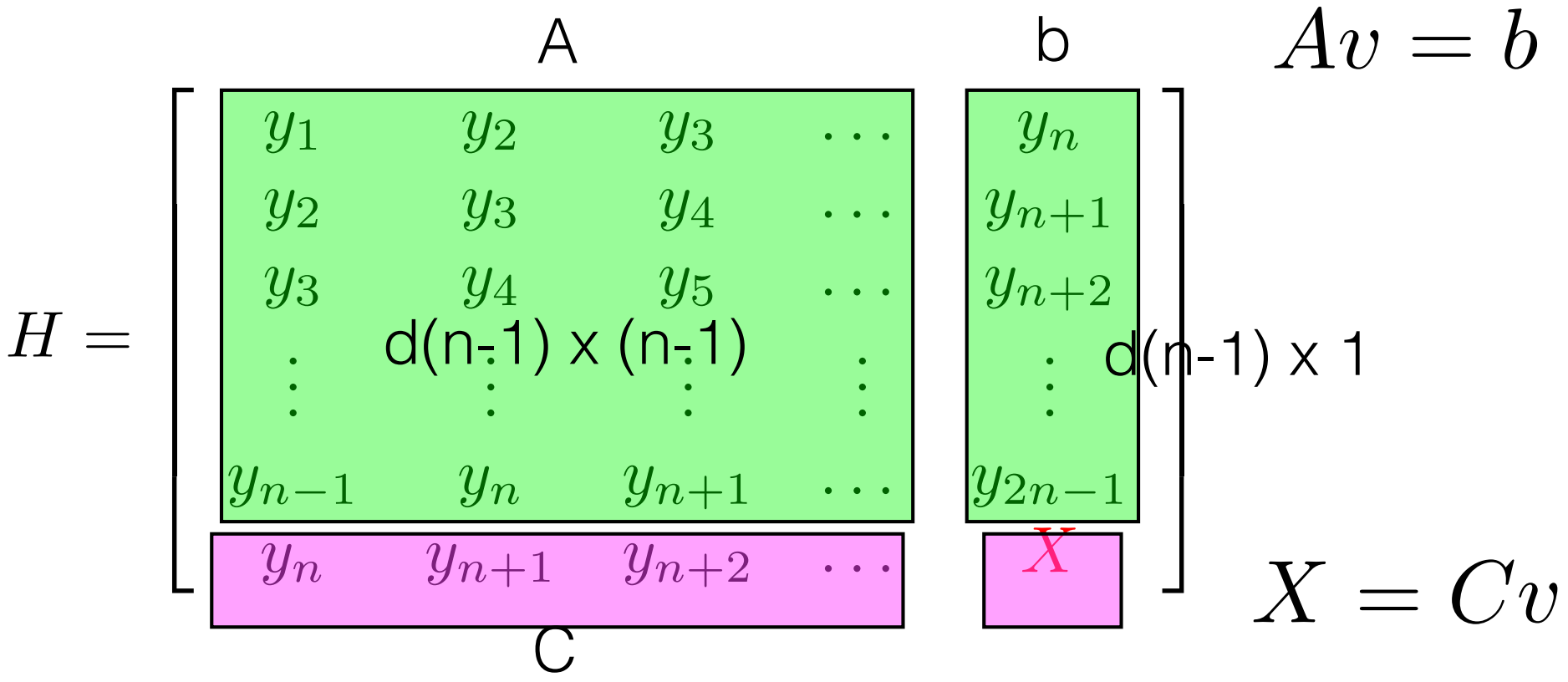
Its Hankel matrix is defined as:

$$H_y = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n-1} & y_n \\ y_2 & y_3 & \cdots & y_n & y_{n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_m & y_{m+1} & \cdots & y_{m+n-2} & y_{m+n-1} \end{bmatrix}$$ md x n

Rank(H_y) measures the complexity of the underlying dynamics: after we have "enough" measurements the rank of the Hankel matrix does not increase.

Predicting the next measurement

The new measurement should not increase the complexity of the dynamics:



The last column must be a linear combination of the previous ones.

Predicting the next measurement

The new measurement should not increase the complexity of the dynamics: $Av = b$

For 2n new measurement, A is d(n-1)x(n-1)

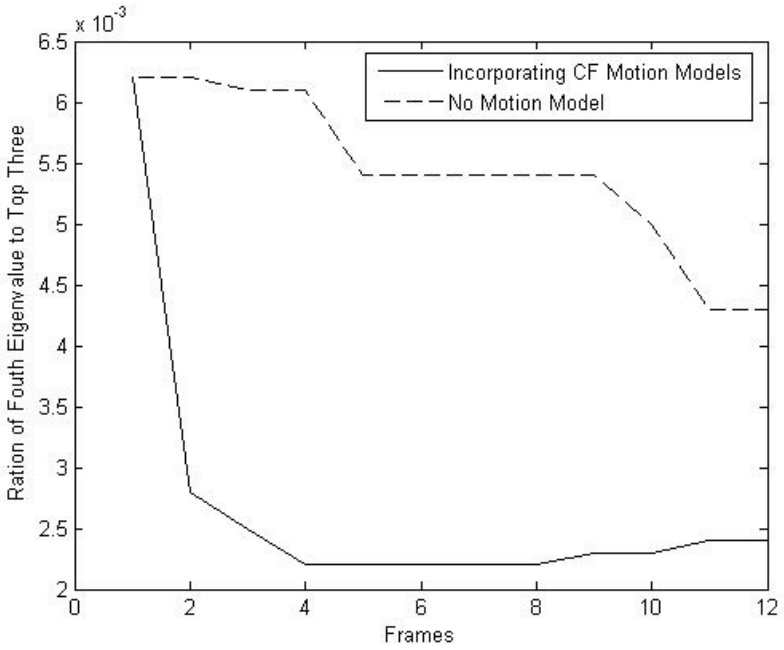
Solve for v using least squares:

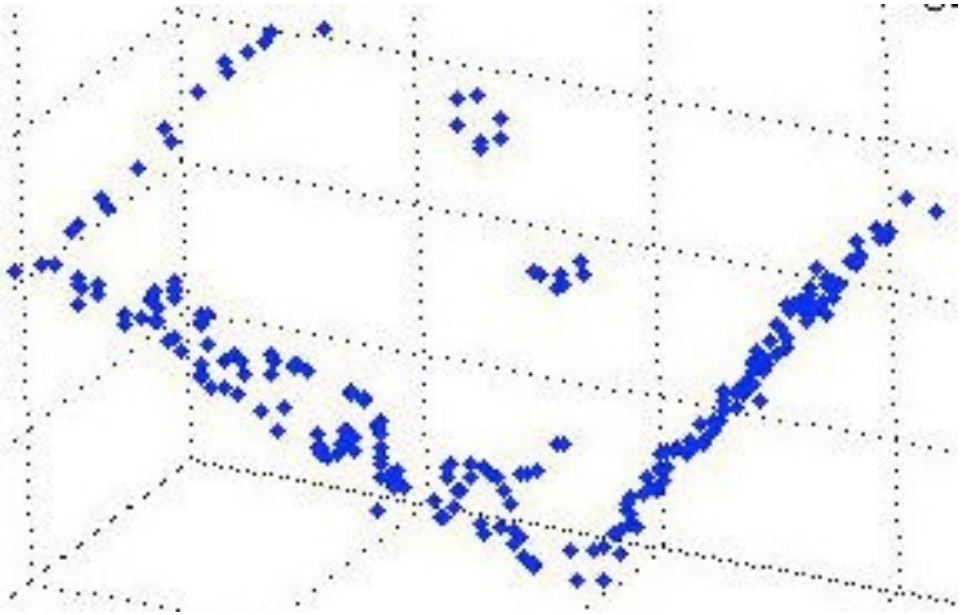
$$v = (A^T A)^{-1} A^T b$$

Use v to compute X:

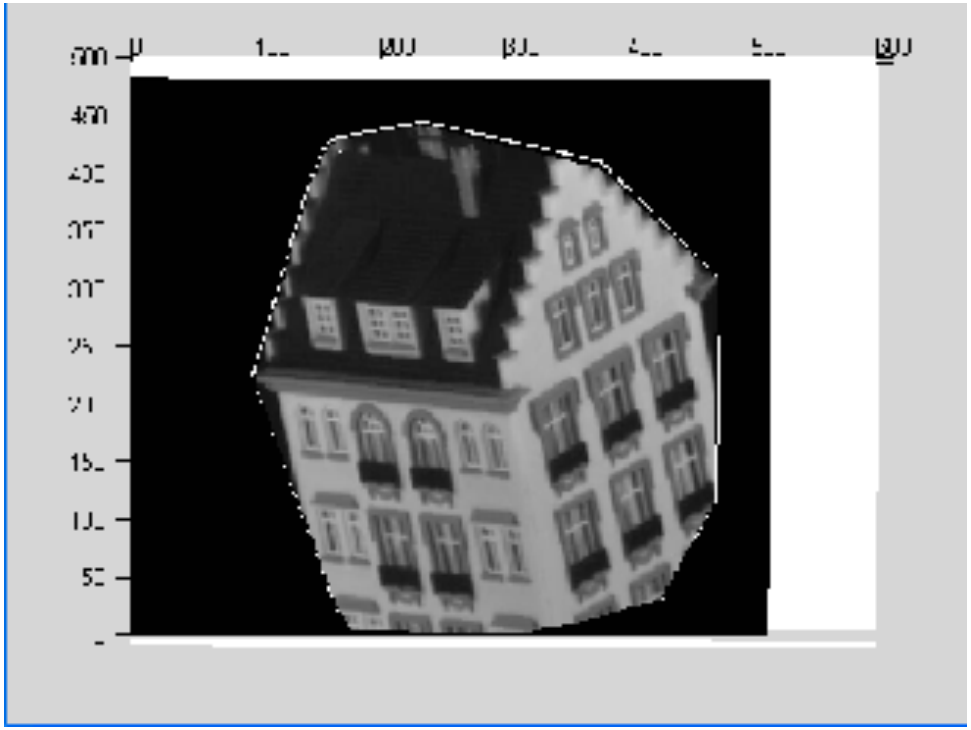
$$X = Cv$$

SfMD


x 10^-3
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
0
2
4
6
8
10
12
Frames
Ration of Fouth Eigenvalue to Top Three
Incorporating CF Motion Models
No Motion Model



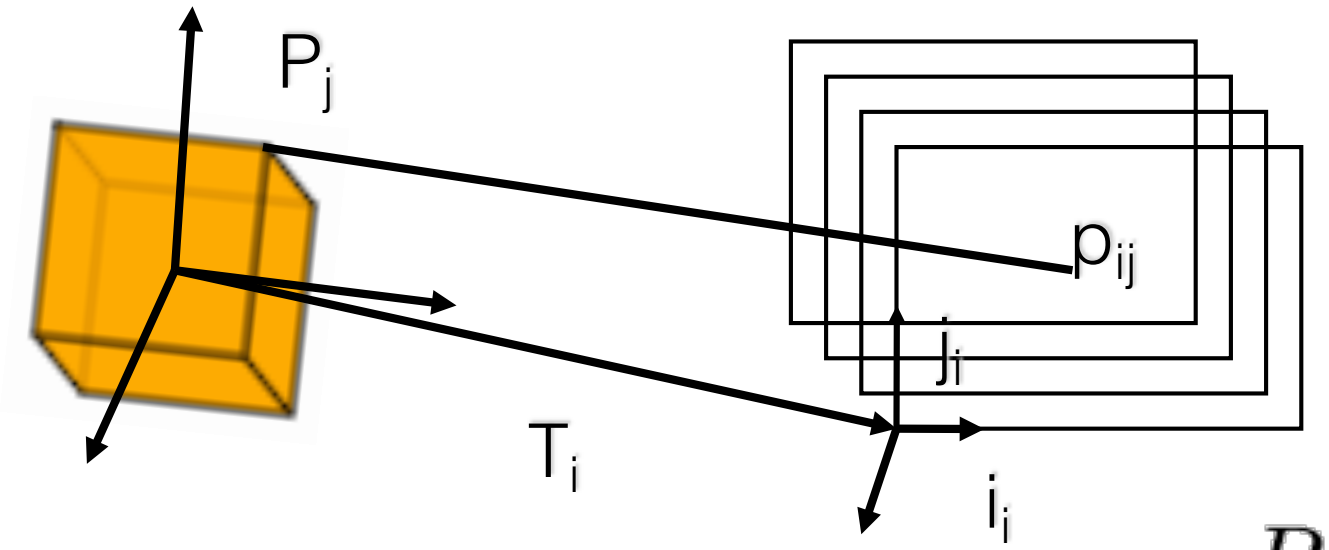
Using Dynamics



SfM Factorization with Dynamics

… back to Tomasi & Kanade Factorization

Orthographic Projection



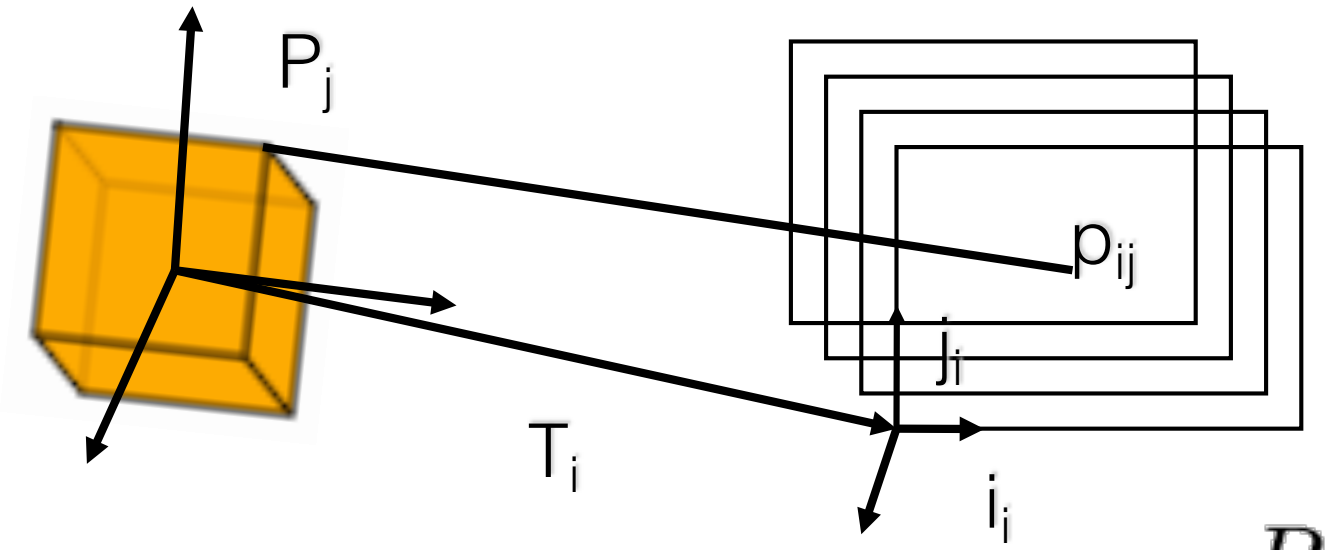
$$p_{ij} = \underset{2x3}{R_j} \underset{3x1}{P_j} + \underset{2x1}{T_i}$$

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix} = \begin{bmatrix} R_1 & T_1 \\ R_2 & T_1 \\ \vdots & \vdots \\ R_m & T_m \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \cdots & P_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

W M S

$$W = MS$$

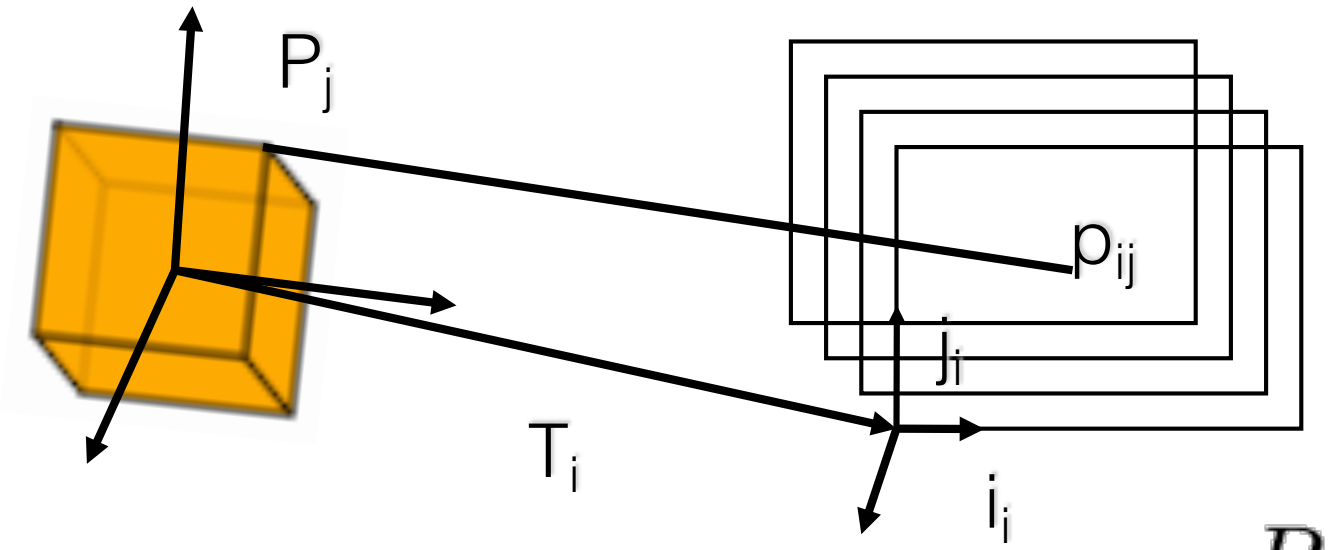
Orthographic Projection



$$p_{ij} = \underset{2x3}{R_j} \underset{3x1}{P_j} + \underset{2x1}{T_i}$$

One can eliminate T by setting the origin of both coordinate systems (the object and the image coordinate systems) at the centroids of all the points (3D and each 2D frame).

Orthographic Projection

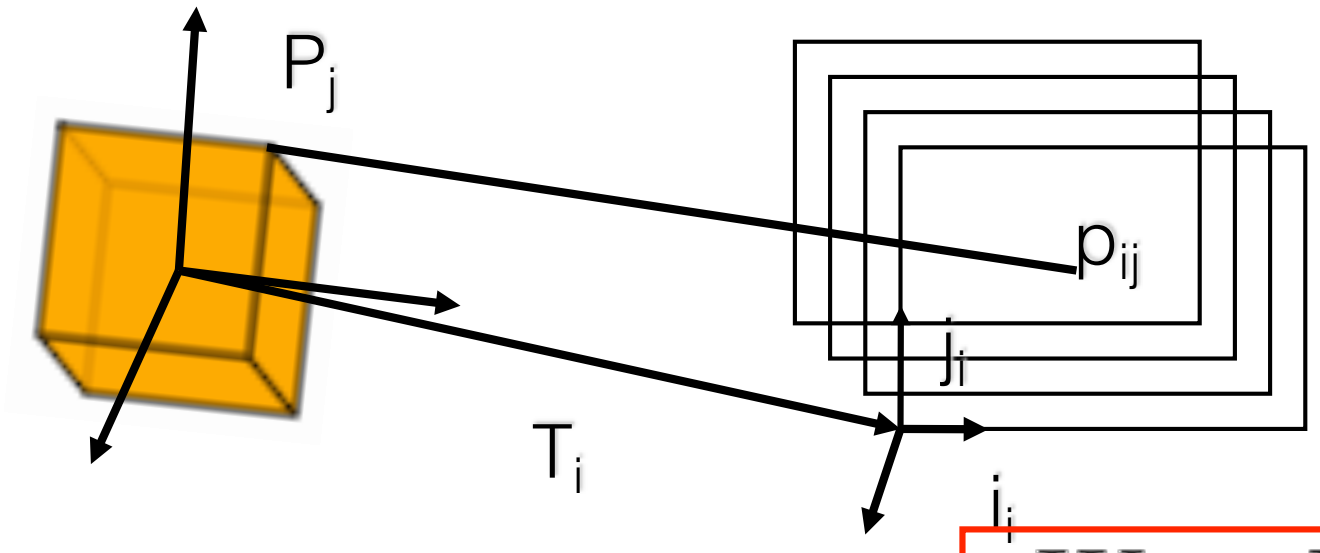


$$p_{ij} = \underset{2x3}{R_j} \underset{3x1}{P_j} + \underset{2x1}{T_i}$$

$$\begin{bmatrix} \hat{p}_{11} & \hat{p}_{12} & \cdots & \hat{p}_{1n} \\ \hat{p}_{21} & \hat{p}_{22} & \cdots & \hat{p}_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{p}_{m1} & \hat{p}_{m2} & \cdots & \hat{p}_{mn} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{bmatrix} \begin{bmatrix} \hat{P}_1 & \hat{P}_2 & \cdots & \hat{P}_n \end{bmatrix}$$

W M S

$$W = MS$$



$$W = MS$$

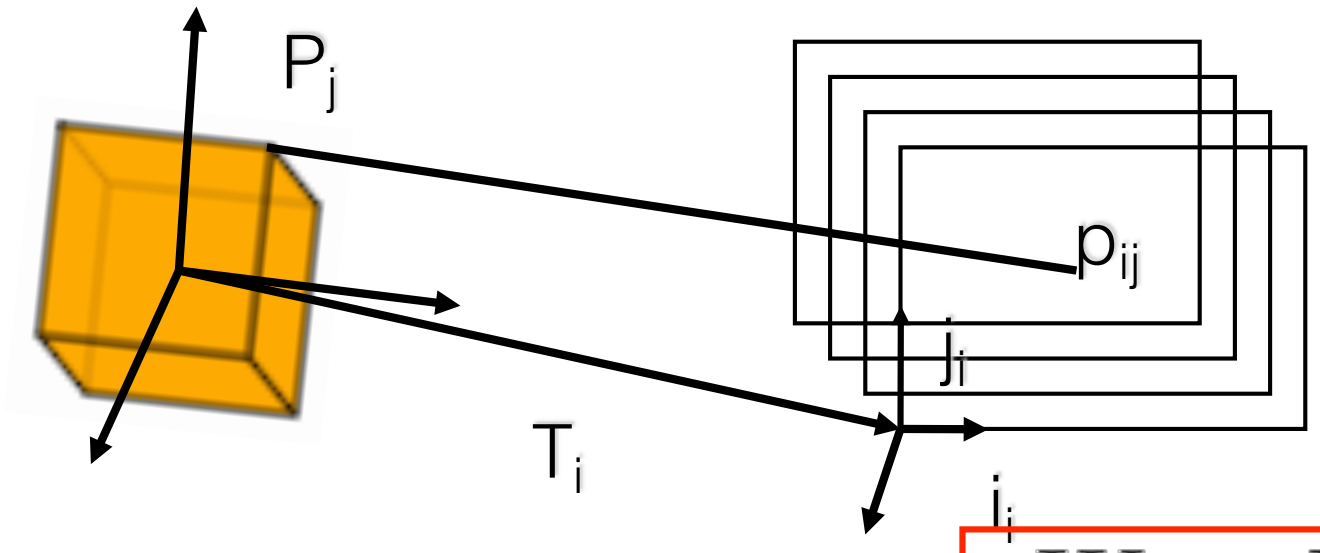
2m x n 2m x 3 3 x n

W is **rank 3** and M and S can be found (up to an affine transf.) from the SVD of W

$$W = UDV^T$$

$$M = UD^{\frac{1}{2}}Q \qquad S = Q^{-1}D^{\frac{1}{2}}V^T$$

Finding Q



$$W = MS$$

2m x n 2m x 3 3 x n

$$W = UDV^T \qquad S = Q^{-1}D^{\frac{1}{2}}V^T$$

$$M = UD^{\frac{1}{2}}Q$$

M should be a stack of ROTATIONS!

Finding Q

$$M = U D^{\frac{1}{2}} Q$$

M should be a stack of ROTATIONS!

$$\begin{aligned} m_{i1}.m_{i2} &= 0 \\ |m_{i1}|^2 &= 1 \\ |m_{i2}|^2 &= 1 \end{aligned}$$

$$m_{i1} \times m_{i2} = m_{i3}$$

Recovering T

$$T_i = R_i \begin{bmatrix} \bar{p}_i \\ \alpha \end{bmatrix} \quad \text{with } \alpha \text{ an arbitrary real number}$$

Algorithm Summary

Obtain the registered measurement matrix W (subtract the mean in each frame)

Compute SVD of $W=UDV^T$

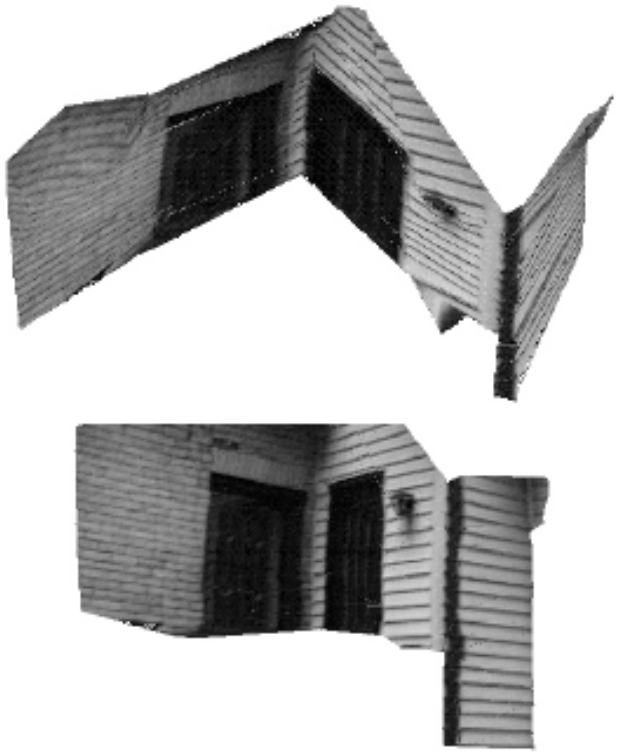
Keep the first 3 eigenvalues/eigenvectors: $W' = U'D'V'^T$

Define $M = U'D'^{1/2}$ and $S = D'^{1/2}V'^T$

Find affinity Q such that MQ is a stack of rotations

Compute T

Reconstruction Results (Tomasi and Kanade, 1992)



Reprinted from Tomasi and Kanade 1992

Further Factorization work

Factorization with uncertainty

Factorization for indep. moving objects (Irani & Anandan, IJCV'02)

Factorization for dynamic objects

Perspective factorization (Costeira and Kanade '94)

Factorization with outliers and missing pts. (Bregler et al. 2000, Brand 2001)

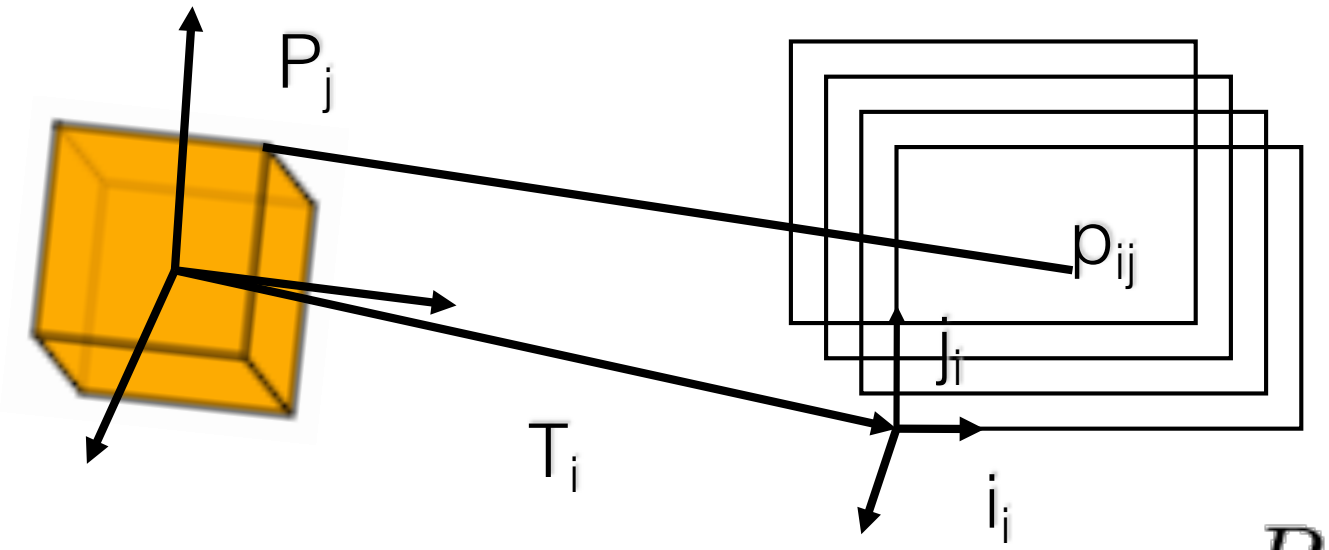
(Sturm & Triggs 1996, Ayazoglu, Sznaier, Camps 2010…)

(Jacobs 1997 (affine),
Martinek and Pajdla 2001,
Aanaes 2002 (perspective))

Motion Based Segmentation by Factorization

Costeira & Kanade ‘98

Motion based Segmentation by Factorization

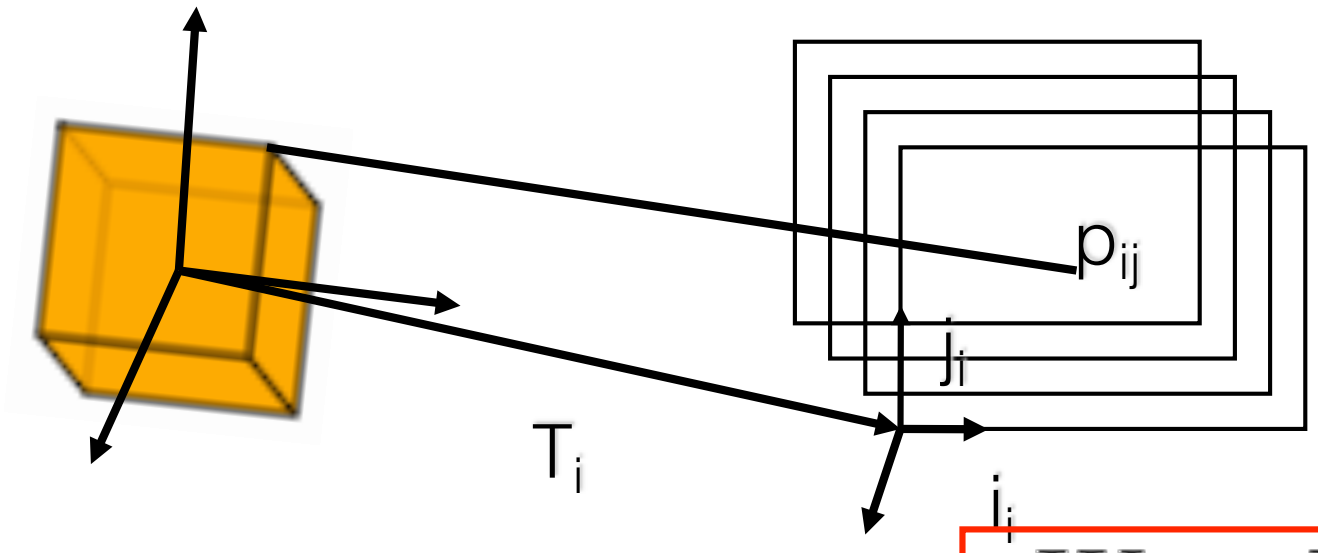


$$p_{ij} = R_i P_j + T_i$$

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix} = \begin{bmatrix} R_1 & T_1 \\ R_2 & T_1 \\ \vdots & \vdots \\ R_m & T_m \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \cdots & P_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

W M S

$$W = MS$$



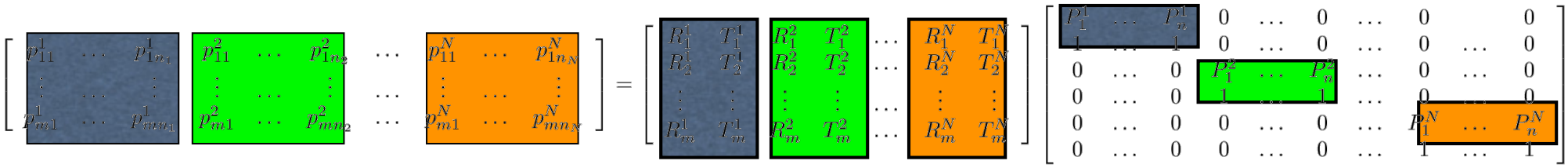
$$W = MS$$

2m x n 2m x 4 4 x n

W has at most rank 4

Motion based Segmentation by Factorization

For N (segmented) rigid objects tracked across m frames:



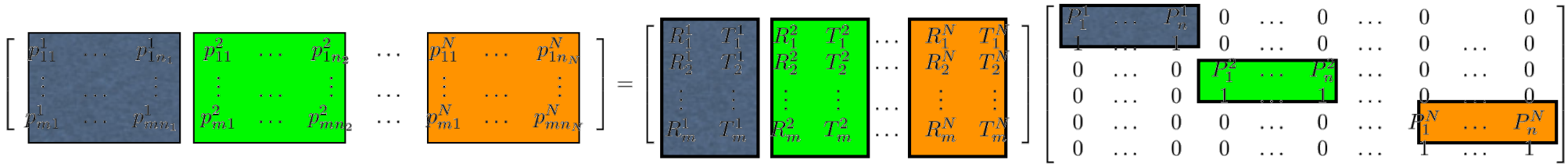
$2m \times (n_1+n_2+\ldots+n_N)$ $\quad 2m \times 4N \quad$ $4N \times (n_1+n_2+\ldots+n_N)$

$$W = MS$$

W has at most rank 4N

Motion based Segmentation by Factorization

For N (segmented) rigid objects tracked across m frames:



$2m \times (n_1+n_2+\ldots+n_N)$ $\quad 2m \times 4N$ $\quad 4N \times (n_1+n_2+\ldots+n_N)$

$$W = MS$$

$$W = UDV^T$$

$$Z = V(V^TV)^{-1}V^T = VV^T$$

Is block diagonal

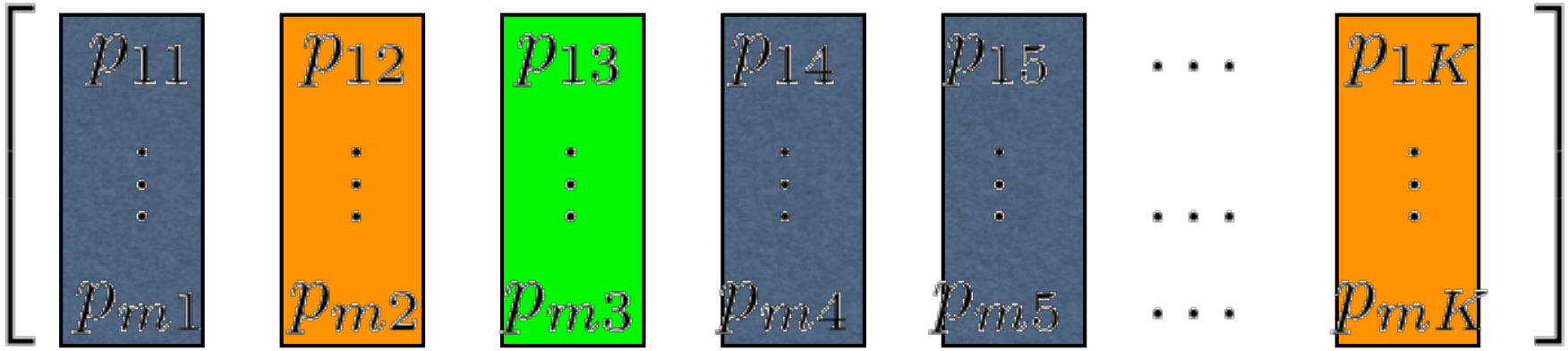
Can we remove the translation component?

No ...

Each object has a different centroid, but there is only one centroid per frame.

Motion based Segmentation by Factorization

For N (unsegmented) rigid objects tracked across m frames:



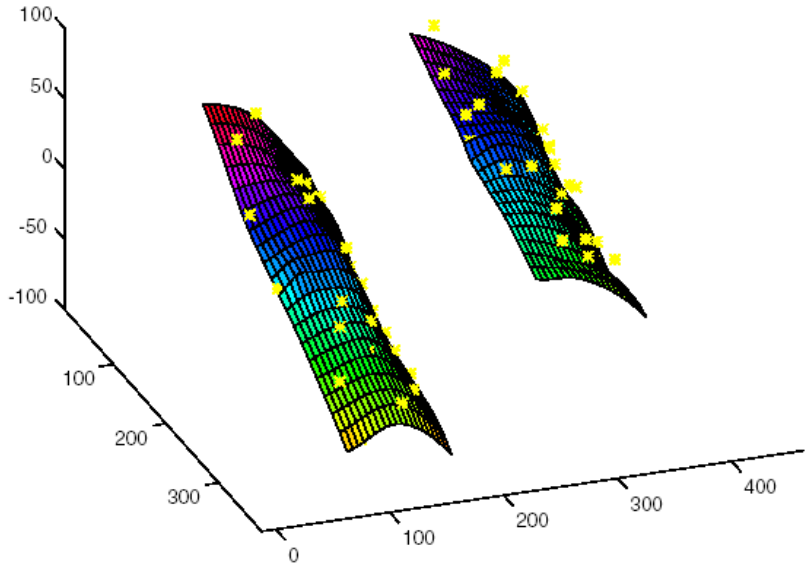
Find SVD of W: $W = UDV^T$

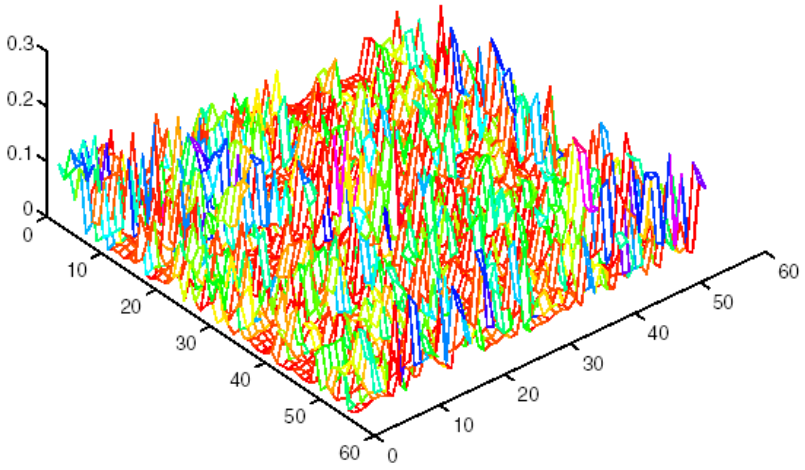
Find "shape interaction matrix":

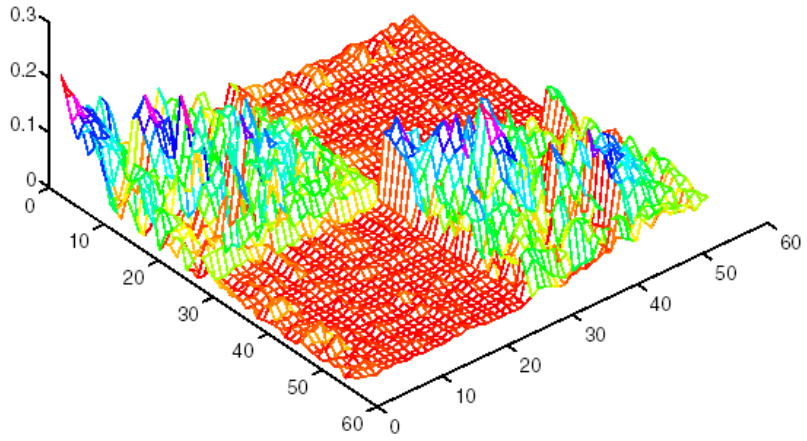
$$Z = VV^T$$

Swap rows and columns of Z until it is block diagonal.

Multiple indep. moving objects







Problems:

Sensitive to noise, outliers, biased tracking …

Cannot handle shared motions modes

Ex:

$$M = \begin{bmatrix} M^1 & M^2 & \dots & M^N \end{bmatrix}$$ With rank 16

How many objects? 16 = 4N ➡ N = 4

But with shared translation:

$$T^1 = T^2 = \dots = T^N = T$$

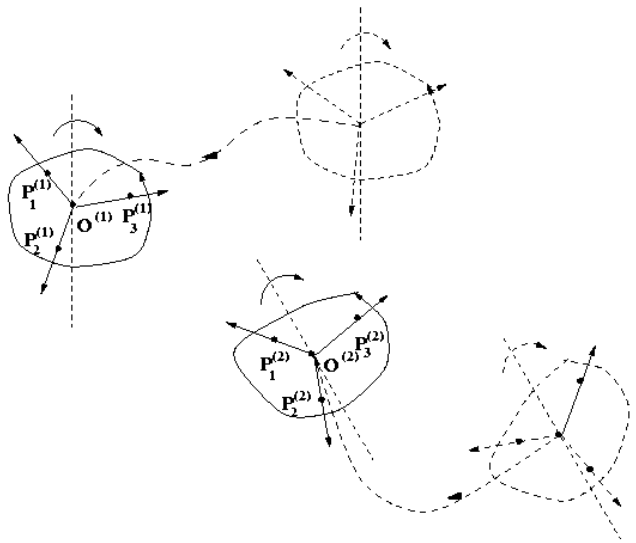
M has at most rank 3N + 1 !!

16 = 3N + 1 ➡ N = 5

Motion Segmentation from Dynamics

Lublinerman, Sznaier, Camps ‘06

Observations:



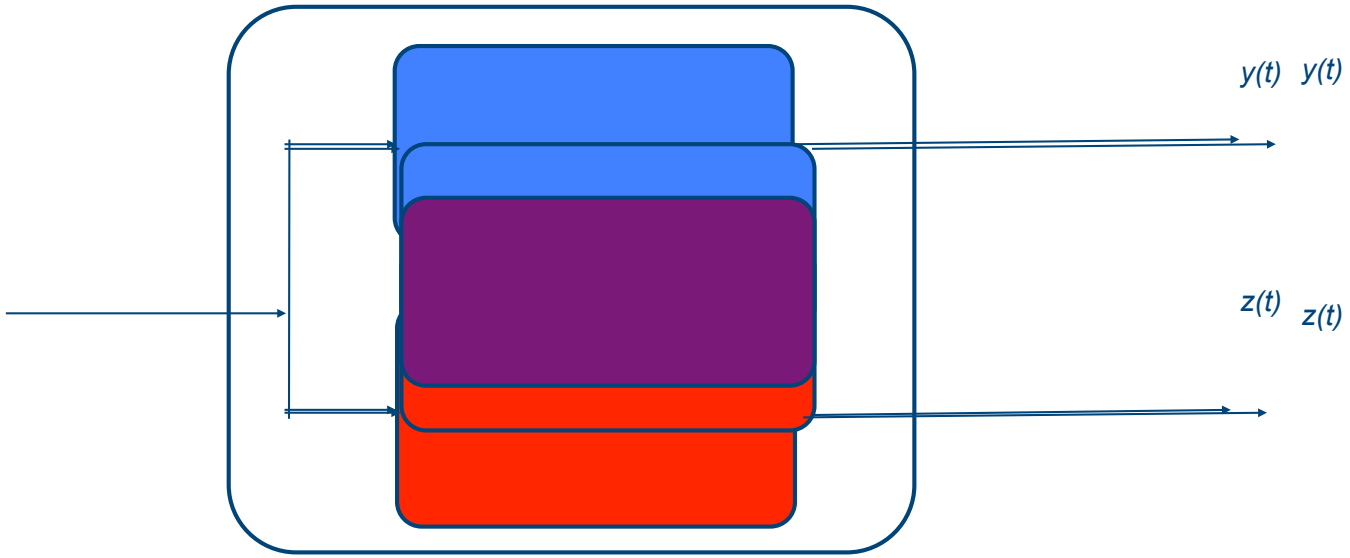
•Temporal information is not used: swapping rows in W does not affect the end result

•The motion of **pairs of points** from a **single object** can be described by the motion of **a** basis attached to the object but does not carry information about the motion of the other objects.

(unobservable states)

•The motion of **pair of points** from **two different objects** can be described by the motion of **two** basis, each of them attached to one of the objects.

Dynamic Data Association



A system explaining independent sequences has higher complexity than a system explaining correlated data.

Look for simplest **joint** models.

(no need to explicitly find the models!)

Main Ideas:

Group points according to the complexity of a linear operator required to explain their spatio-temporal evolution.

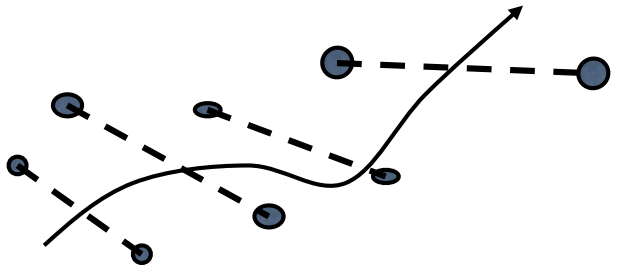
The order of the operator can be estimated by computing the rank of its Hankel matrix.

Dynamics based Segmentation

$$W = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_1 \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix}$$

n points, m frames

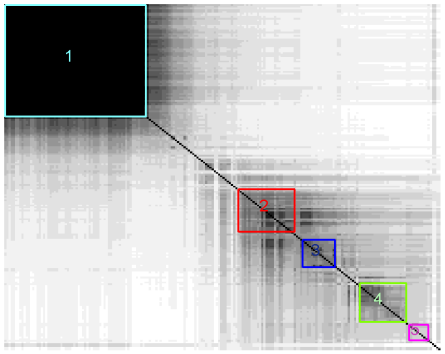
Relative distance between $\mathbf{p}_i$ and $\mathbf{p}_j$:



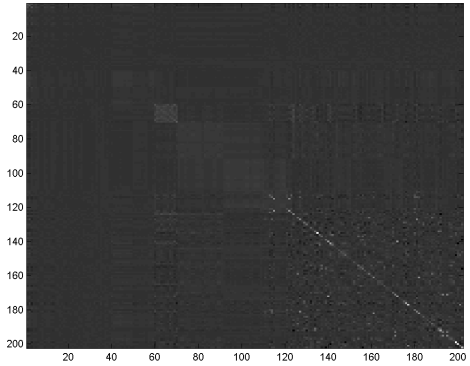
$$d_t^{i,j} = p_{ti} - p_{tj}$$

$$H^{i,j} = \begin{bmatrix} d_1^{i,j} & d_2^{i,j} & \cdots & d_{m/2}^{i,j} \\ d_2^{i,j} & d_3^{i,j} & \cdots & d_{m/2+1}^{i,j} \\ \vdots & \vdots & \vdots & \vdots \\ d_{m/2}^{i,j} & \cdots & \cdots & d_m^{i,j} \end{bmatrix}$$

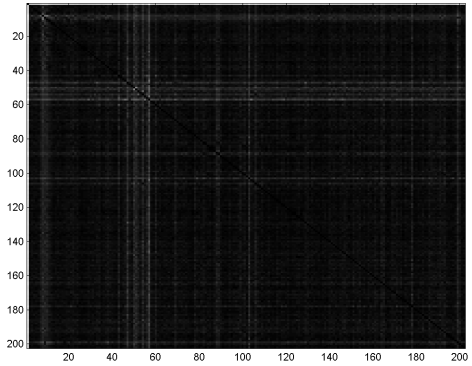
If pi and pj move together, the order of the dynamics needed to explain their motion has lower order than when they move independently.



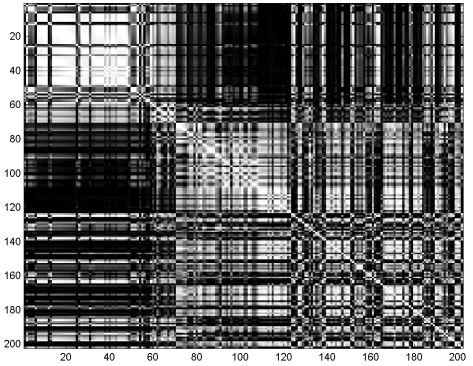
Hankel



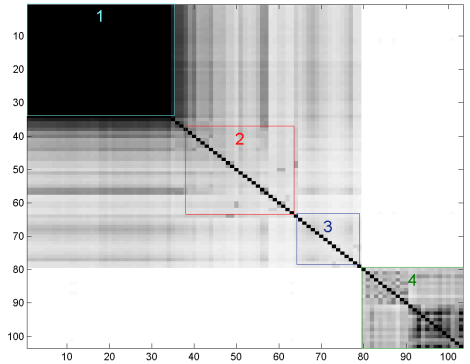
Costeira Kanade



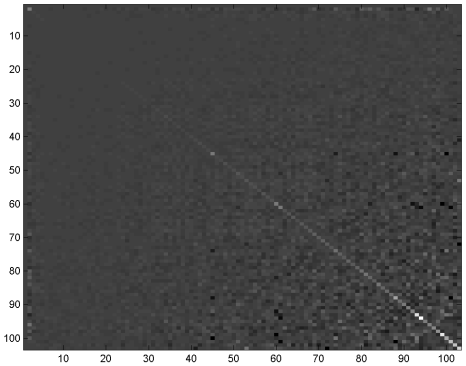
Irani



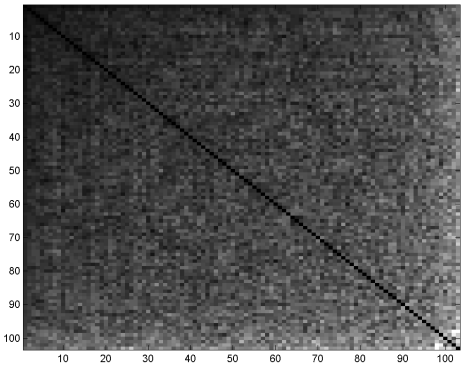
GPCA



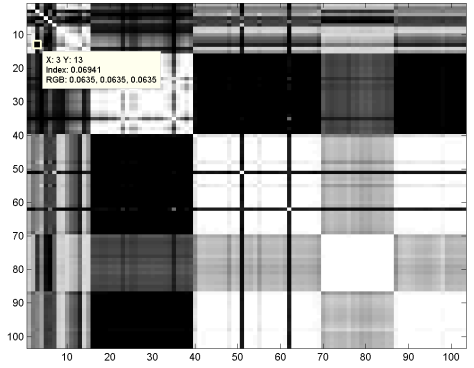
Hankel



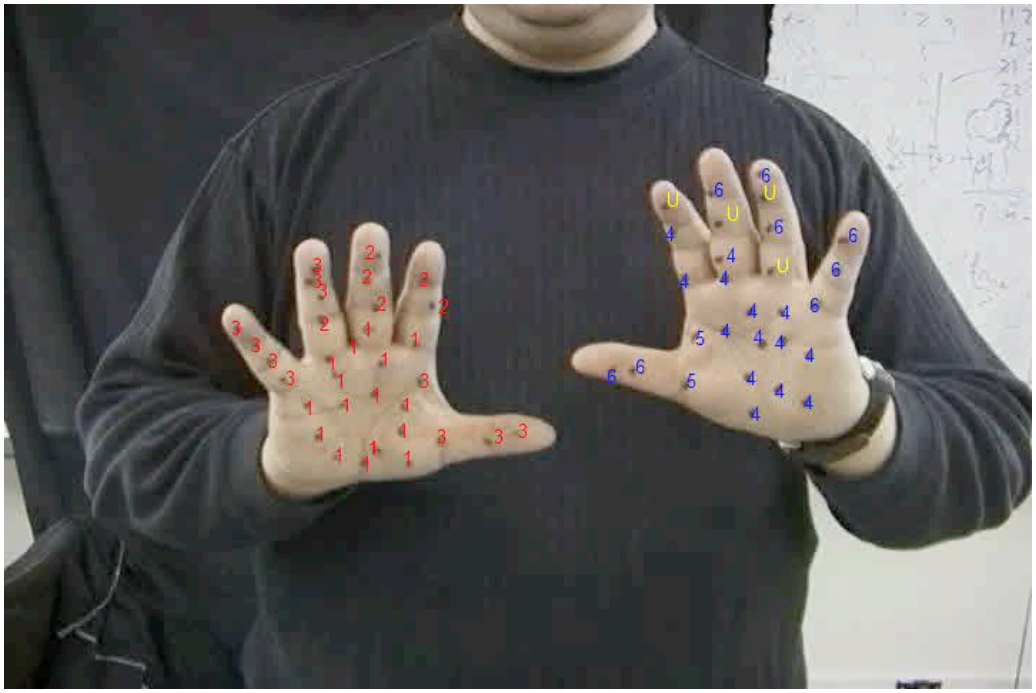
Costeira Kanade

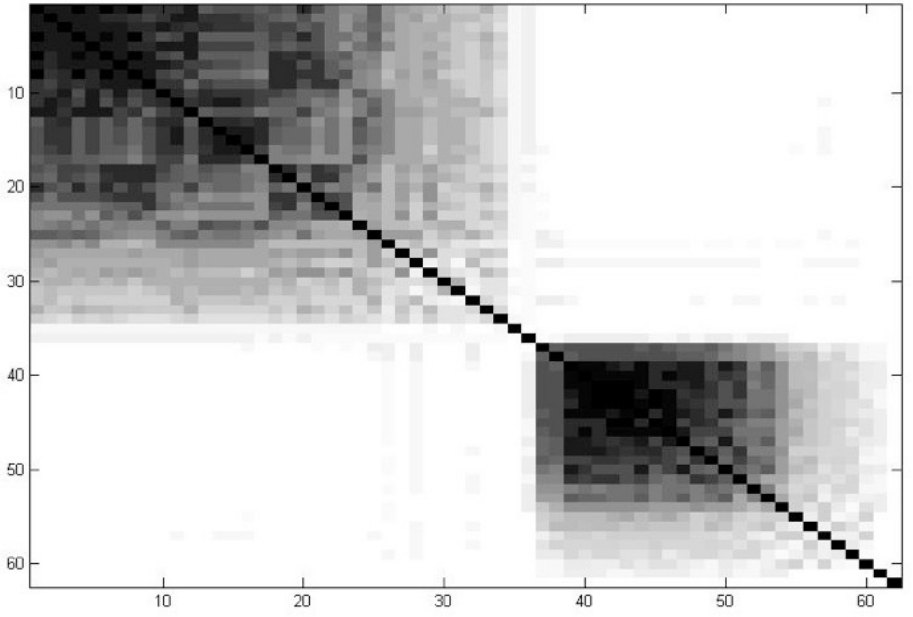


Irani



GPCA




10
20
30
40
50
60
10
20
30
40
50
60